# Business Objectives 🎯

This case study aims to identify patterns which indicate if a client has difficulty paying their instalments which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc. This will ensure that the consumers capable of repaying the loan are not rejected. Identification of such applicants using EDA is the aim of this case study.

In other words, the company wants to understand **the driving factors (or driver variables) behind loan default, i.e. the variables which are strong indicators of default.** The company can utilise this knowledge for its portfolio and risk assessment.

Import the libraries.

```
# This is formatted as code

# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

pd.set_option('display.max_columns',125)
pd.set_option('display.max_rows',200)
```

Read the Data set

```
# load application_data file
import pandas as pd

# Load application_data file
application_data = pd.read_csv('application_data.csv')
application_data.head()

{"type":"dataframe","variable_name":"application_data"}
```

Check structure of data

```
# check structure of data
print(application_data.shape)

(307511, 81)

print(application_data.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 81 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   SK_ID_CURR                    307511 non-null  int64
 1   TARGET                        307511 non-null  object
 2   NAME_CONTRACT_TYPE            307511 non-null  object
 3   CODE_GENDER                   307511 non-null  object
 4   FLAG_OWN_CAR                  307511 non-null  object
 5   FLAG_OWN_REALTY               307511 non-null  object
 6   CNT_CHILDREN                  307511 non-null  int64
 7   AMT_INCOME_TOTAL              307511 non-null  float64
 8   AMT_CREDIT                    307511 non-null  float64
 9   AMT_ANNUITY                   307499 non-null  float64
 10  AMT_GOODS_PRICE               307233 non-null  float64
 11  NAME_TYPE_SUITE               306219 non-null  object
 12  NAME_INCOME_TYPE              307511 non-null  object
 13  NAME_EDUCATION_TYPE           307511 non-null  object
 14  NAME_FAMILY_STATUS            307511 non-null  object
 15  NAME_HOUSING_TYPE             307511 non-null  object
 16  REGION_POPULATION_RELATIVE    307511 non-null  float64
 17  DAYS_BIRTH                    307511 non-null  int64
 18  DAYS_EMPLOYED                 307511 non-null  int64
 19  DAYS_REGISTRATION             307511 non-null  float64
 20  DAYS_ID_PUBLISH               307511 non-null  int64
 21  FLAG_MOBIL                    307511 non-null  object
 22  FLAG_EMP_PHONE                307511 non-null  object
 23  FLAG_WORK_PHONE               307511 non-null  object
 24  FLAG_CONT_MOBILE              307511 non-null  object
 25  FLAG_PHONE                    307511 non-null  object
 26  FLAG_EMAIL                    307511 non-null  object
 27  OCCUPATION_TYPE               211120 non-null  object
 28  CNT_FAM_MEMBERS               307509 non-null  float64
 29  REGION_RATING_CLIENT          307511 non-null  object
 30  REGION_RATING_CLIENT_W_CITY   307511 non-null  object
 31  WEEKDAY_APPR_PROCESS_START    307511 non-null  object
 32  HOUR_APPR_PROCESS_START       307511 non-null  int64
 33  REG_REGION_NOT_LIVE_REGION    307511 non-null  object
 34  REG_REGION_NOT_WORK_REGION    307511 non-null  object
 35  LIVE_REGION_NOT_WORK_REGION   307511 non-null  object
 36  REG_CITY_NOT_LIVE_CITY        307511 non-null  object
 37  REG_CITY_NOT_WORK_CITY        307511 non-null  object
 38  LIVE_CITY_NOT_WORK_CITY       307511 non-null  object
 39  ORGANIZATION_TYPE             307511 non-null  object
 40  EXT_SOURCE_2                  306851 non-null  float64
 41  EXT_SOURCE_3                  246546 non-null  float64
 42  YEARS_BEGINEXPLUATATION_AVG   157504 non-null  float64
 43  FLOORSMAX_AVG                 154491 non-null  float64
 44  YEARS_BEGINEXPLUATATION_MODE  157504 non-null  float64
 45  FLOORSMAX_MODE                154491 non-null  float64
 46  YEARS_BEGINEXPLUATATION_MEDI  157504 non-null  float64
 47  FLOORSMAX_MEDI                  154491 non-null   float64
 48  TOTALAREA_MODE                  159080 non-null   float64
 49  EMERGENCYSTATE_MODE             161756 non-null   object 
 50  OBS_30_CNT_SOCIAL_CIRCLE        306490 non-null   float64
 51  DEF_30_CNT_SOCIAL_CIRCLE        306490 non-null   float64
 52  OBS_60_CNT_SOCIAL_CIRCLE        306490 non-null   float64
 53  DEF_60_CNT_SOCIAL_CIRCLE        306490 non-null   float64
 54  DAYS_LAST_PHONE_CHANGE          307510 non-null   float64
 55  FLAG_DOCUMENT_2                 307511 non-null   object 
 56  FLAG_DOCUMENT_3                 307511 non-null   object 
 57  FLAG_DOCUMENT_4                 307511 non-null   object 
 58  FLAG_DOCUMENT_5                 307511 non-null   object 
 59  FLAG_DOCUMENT_6                 307511 non-null   object 
 60  FLAG_DOCUMENT_7                 307511 non-null   object 
 61  FLAG_DOCUMENT_8                 307511 non-null   object 
 62  FLAG_DOCUMENT_9                 307511 non-null   object 
 63  FLAG_DOCUMENT_10                307511 non-null   object 
 64  FLAG_DOCUMENT_11                307511 non-null   object 
 65  FLAG_DOCUMENT_12                307511 non-null   object 
 66  FLAG_DOCUMENT_13                307511 non-null   object 
 67  FLAG_DOCUMENT_14                307511 non-null   object 
 68  FLAG_DOCUMENT_15                307511 non-null   object 
 69  FLAG_DOCUMENT_16                307511 non-null   object 
 70  FLAG_DOCUMENT_17                307511 non-null   object 
 71  FLAG_DOCUMENT_18                307511 non-null   object 
 72  FLAG_DOCUMENT_19                307511 non-null   object 
 73  FLAG_DOCUMENT_20                307511 non-null   object 
 74  FLAG_DOCUMENT_21                307511 non-null   object 
 75  AMT_REQ_CREDIT_BUREAU_HOUR      265992 non-null   float64
 76  AMT_REQ_CREDIT_BUREAU_DAY       265992 non-null   float64
 77  AMT_REQ_CREDIT_BUREAU_WEEK      265992 non-null   float64
 78  AMT_REQ_CREDIT_BUREAU_MON       265992 non-null   float64
 79  AMT_REQ_CREDIT_BUREAU_QRT       265992 non-null   float64
 80  AMT_REQ_CREDIT_BUREAU_YEAR      265992 non-null   float64
dtypes: float64(27), int64(6), object(48)
memory usage: 190.0+ MB
None
```

```
application_data.describe()
```

{"type":"dataframe"}

Check Data quality and missing values

```
#find the percentage of missing values for all the columns
round(100*application_data.isnull().sum()/len(application_data),2)
```

```
SK_ID_CURR                      0.00
TARGET                          0.05
NAME_CONTRACT_TYPE              0.05
```

```
CODE_GENDER                      0.05
FLAG_OWN_CAR                     0.05
FLAG_OWN_REALTY                  0.05
CNT_CHILDREN                     0.05
AMT_INCOME_TOTAL                 0.05
AMT_CREDIT                       0.05
AMT_ANNUITY                      0.05
AMT_GOODS_PRICE                  0.10
NAME_TYPE_SUITE                  0.31
NAME_INCOME_TYPE                 0.05
NAME_EDUCATION_TYPE              0.05
NAME_FAMILY_STATUS               0.05
NAME_HOUSING_TYPE                0.05
REGION_POPULATION_RELATIVE       0.05
DAYS_BIRTH                       0.05
DAYS_EMPLOYED                    0.05
DAYS_REGISTRATION                0.05
DAYS_ID_PUBLISH                  0.05
OWN_CAR_AGE                     65.26
FLAG_MOBIL                       0.05
FLAG_EMP_PHONE                   0.05
FLAG_WORK_PHONE                  0.05
FLAG_CONT_MOBILE                 0.05
FLAG_PHONE                       0.05
FLAG_EMAIL                       0.05
OCCUPATION_TYPE                 29.33
CNT_FAM_MEMBERS                  0.05
REGION_RATING_CLIENT             0.05
REGION_RATING_CLIENT_W_CITY      0.05
WEEKDAY_APPR_PROCESS_START       0.05
HOUR_APPR_PROCESS_START          0.05
REG_REGION_NOT_LIVE_REGION       0.05
REG_REGION_NOT_WORK_REGION       0.05
LIVE_REGION_NOT_WORK_REGION      0.05
REG_CITY_NOT_LIVE_CITY           0.05
REG_CITY_NOT_WORK_CITY           0.05
LIVE_CITY_NOT_WORK_CITY          0.05
ORGANIZATION_TYPE                0.05
EXT_SOURCE_1                    56.37
EXT_SOURCE_2                     0.31
EXT_SOURCE_3                    20.85
APARTMENTS_AVG                  49.77
BASEMENTAREA_AVG                57.25
YEARS_BEGINEXPLUATATION_AVG     47.95
YEARS_BUILD_AVG                 64.74
COMMONAREA_AVG                  68.59
ELEVATORS_AVG                   51.95
ENTRANCES_AVG                   48.99
FLOORSMAX_AVG                   48.26
```

```
FLOORSMIN_AVG                   66.67
LANDAREA_AVG                    58.19
LIVINGAPARTMENTS_AVG            67.24
LIVINGAREA_AVG                  49.04
NONLIVINGAPARTMENTS_AVG         68.17
NONLIVINGAREA_AVG               53.25
APARTMENTS_MODE                 49.77
BASEMENTAREA_MODE               57.25
YEARS_BEGINEXPLUATATION_MODE    47.95
YEARS_BUILD_MODE                64.74
COMMONAREA_MODE                 68.59
ELEVATORS_MODE                  51.95
ENTRANCES_MODE                  48.99
FLOORSMAX_MODE                  48.26
FLOORSMIN_MODE                  66.67
LANDAREA_MODE                   58.19
LIVINGAPARTMENTS_MODE           67.24
LIVINGAREA_MODE                 49.04
NONLIVINGAPARTMENTS_MODE        68.17
NONLIVINGAREA_MODE              53.25
APARTMENTS_MEDI                 49.77
BASEMENTAREA_MEDI               57.25
YEARS_BEGINEXPLUATATION_MEDI    47.95
YEARS_BUILD_MEDI                64.74
COMMONAREA_MEDI                 68.59
ELEVATORS_MEDI                  51.95
ENTRANCES_MEDI                  48.99
FLOORSMAX_MEDI                  48.26
FLOORSMIN_MEDI                  66.67
LANDAREA_MEDI                   58.19
LIVINGAPARTMENTS_MEDI           67.24
LIVINGAREA_MEDI                 49.04
NONLIVINGAPARTMENTS_MEDI        68.17
NONLIVINGAREA_MEDI              53.25
FONDKAPREMONT_MODE              66.61
HOUSETYPE_MODE                  48.62
TOTALAREA_MODE                  47.27
WALLSMATERIAL_MODE              49.45
EMERGENCYSTATE_MODE             46.18
OBS_30_CNT_SOCIAL_CIRCLE         0.62
DEF_30_CNT_SOCIAL_CIRCLE         0.62
OBS_60_CNT_SOCIAL_CIRCLE         0.62
DEF_60_CNT_SOCIAL_CIRCLE         0.62
DAYS_LAST_PHONE_CHANGE           0.05
FLAG_DOCUMENT_2                  0.05
FLAG_DOCUMENT_3                  0.05
FLAG_DOCUMENT_4                  0.05
FLAG_DOCUMENT_5                  0.05
FLAG_DOCUMENT_6                  0.05
```

```
FLAG_DOCUMENT_7                  0.05
FLAG_DOCUMENT_8                  0.05
FLAG_DOCUMENT_9                  0.05
FLAG_DOCUMENT_10                 0.05
FLAG_DOCUMENT_11                 0.05
FLAG_DOCUMENT_12                 0.05
FLAG_DOCUMENT_13                 0.05
FLAG_DOCUMENT_14                 0.05
FLAG_DOCUMENT_15                 0.05
FLAG_DOCUMENT_16                 0.05
FLAG_DOCUMENT_17                 0.05
FLAG_DOCUMENT_18                 0.05
FLAG_DOCUMENT_19                 0.05
FLAG_DOCUMENT_20                 0.05
FLAG_DOCUMENT_21                 0.05
AMT_REQ_CREDIT_BUREAU_HOUR      14.61
AMT_REQ_CREDIT_BUREAU_DAY       14.61
AMT_REQ_CREDIT_BUREAU_WEEK      14.61
AMT_REQ_CREDIT_BUREAU_MON       14.61
AMT_REQ_CREDIT_BUREAU_QRT       14.61
AMT_REQ_CREDIT_BUREAU_YEAR      14.61
dtype: float64
```

```
# remove columns with high missing percentage
# considering 50% as the threshold value
application_data= application_data.loc[:,
100*application_data.isnull().sum()/len(application_data) < 50]
# checking for shape of the data
application_data.shape
```

```
(307511, 81)
```

For columns with a lower percentage of missing values (approximately 13% or less), determining the optimal metric for imputing missing values is crucial. For categorical columns, explore which category could be used to fill the null values. For numerical columns, assess whether mean or median imputation is appropriate. In some cases, filling missing values with 0 might be suitable. This task should be conducted selectively for a subset of variables, typically around 5-6, rather than all columns.

```
# checking for percentage of null values
round(100*application_data.isnull().sum()/len(application_data),2)
```

```
SK_ID_CURR                       0.00
TARGET                           0.00
NAME_CONTRACT_TYPE               0.00
CODE_GENDER                      0.00
FLAG_OWN_CAR                     0.00
FLAG_OWN_REALTY                  0.00
CNT_CHILDREN                     0.00
```

```
AMT_INCOME_TOTAL                 0.00
AMT_CREDIT                       0.00
AMT_ANNUITY                      0.00
AMT_GOODS_PRICE                  0.09
NAME_TYPE_SUITE                  0.42
NAME_INCOME_TYPE                 0.00
NAME_EDUCATION_TYPE              0.00
NAME_FAMILY_STATUS               0.00
NAME_HOUSING_TYPE                0.00
REGION_POPULATION_RELATIVE       0.00
DAYS_BIRTH                       0.00
DAYS_EMPLOYED                    0.00
DAYS_REGISTRATION                0.00
DAYS_ID_PUBLISH                  0.00
FLAG_MOBIL                       0.00
FLAG_EMP_PHONE                   0.00
FLAG_WORK_PHONE                  0.00
FLAG_CONT_MOBILE                 0.00
FLAG_PHONE                       0.00
FLAG_EMAIL                       0.00
OCCUPATION_TYPE                 31.35
CNT_FAM_MEMBERS                  0.00
REGION_RATING_CLIENT             0.00
REGION_RATING_CLIENT_W_CITY      0.00
WEEKDAY_APPR_PROCESS_START       0.00
HOUR_APPR_PROCESS_START          0.00
REG_REGION_NOT_LIVE_REGION       0.00
REG_REGION_NOT_WORK_REGION       0.00
LIVE_REGION_NOT_WORK_REGION      0.00
REG_CITY_NOT_LIVE_CITY           0.00
REG_CITY_NOT_WORK_CITY           0.00
LIVE_CITY_NOT_WORK_CITY          0.00
ORGANIZATION_TYPE                0.00
EXT_SOURCE_2                     0.21
EXT_SOURCE_3                    19.83
YEARS_BEGINEXPLUATATION_AVG     48.78
FLOORSMAX_AVG                   49.76
YEARS_BEGINEXPLUATATION_MODE    48.78
FLOORSMAX_MODE                  49.76
YEARS_BEGINEXPLUATATION_MEDI    48.78
FLOORSMAX_MEDI                  49.76
TOTALAREA_MODE                  48.27
EMERGENCYSTATE_MODE             47.40
OBS_30_CNT_SOCIAL_CIRCLE         0.33
DEF_30_CNT_SOCIAL_CIRCLE         0.33
OBS_60_CNT_SOCIAL_CIRCLE         0.33
DEF_60_CNT_SOCIAL_CIRCLE         0.33
DAYS_LAST_PHONE_CHANGE           0.00
FLAG_DOCUMENT_2                  0.00
```

```
FLAG_DOCUMENT_3                  0.00
FLAG_DOCUMENT_4                  0.00
FLAG_DOCUMENT_5                  0.00
FLAG_DOCUMENT_6                  0.00
FLAG_DOCUMENT_7                  0.00
FLAG_DOCUMENT_8                  0.00
FLAG_DOCUMENT_9                  0.00
FLAG_DOCUMENT_10                 0.00
FLAG_DOCUMENT_11                 0.00
FLAG_DOCUMENT_12                 0.00
FLAG_DOCUMENT_13                 0.00
FLAG_DOCUMENT_14                 0.00
FLAG_DOCUMENT_15                 0.00
FLAG_DOCUMENT_16                 0.00
FLAG_DOCUMENT_17                 0.00
FLAG_DOCUMENT_18                 0.00
FLAG_DOCUMENT_19                 0.00
FLAG_DOCUMENT_20                 0.00
FLAG_DOCUMENT_21                 0.00
AMT_REQ_CREDIT_BUREAU_HOUR      13.50
AMT_REQ_CREDIT_BUREAU_DAY       13.50
AMT_REQ_CREDIT_BUREAU_WEEK      13.50
AMT_REQ_CREDIT_BUREAU_MON       13.50
AMT_REQ_CREDIT_BUREAU_QRT       13.50
AMT_REQ_CREDIT_BUREAU_YEAR      13.50
dtype: float64
```

```
# retriving the columns which has any null values
application_data_columns=application_data.columns[application_data.isn
ull().any()].tolist()
application_data[application_data_columns].isnull().sum()*100/len(appl
ication_data)
```

```
AMT_ANNUITY                      0.003902
AMT_GOODS_PRICE                  0.090403
NAME_TYPE_SUITE                  0.420148
OCCUPATION_TYPE                 31.345545
CNT_FAM_MEMBERS                  0.000650
EXT_SOURCE_2                     0.214626
EXT_SOURCE_3                    19.825307
YEARS_BEGINEXPLUATATION_AVG     48.781019
FLOORSMAX_AVG                   49.760822
YEARS_BEGINEXPLUATATION_MODE    48.781019
FLOORSMAX_MODE                  49.760822
YEARS_BEGINEXPLUATATION_MEDI    48.781019
FLOORSMAX_MEDI                  49.760822
TOTALAREA_MODE                  48.268517
EMERGENCYSTATE_MODE             47.398304
OBS_30_CNT_SOCIAL_CIRCLE         0.332021
DEF_30_CNT_SOCIAL_CIRCLE         0.332021
```

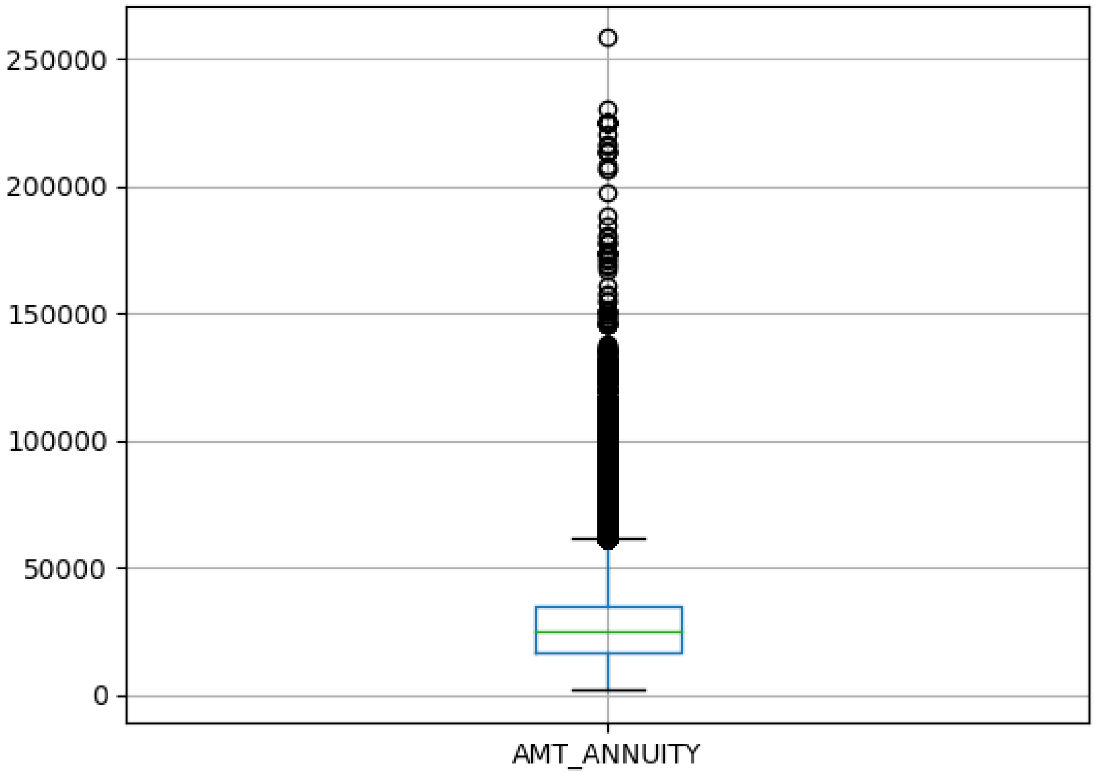
```
OBS_60_CNT_SOCIAL_CIRCLE          0.332021
DEF_60_CNT_SOCIAL_CIRCLE          0.332021
DAYS_LAST_PHONE_CHANGE            0.000325
AMT_REQ_CREDIT_BUREAU_HOUR       13.501631
AMT_REQ_CREDIT_BUREAU_DAY        13.501631
AMT_REQ_CREDIT_BUREAU_WEEK       13.501631
AMT_REQ_CREDIT_BUREAU_MON        13.501631
AMT_REQ_CREDIT_BUREAU_QRT        13.501631
AMT_REQ_CREDIT_BUREAU_YEAR       13.501631
dtype: float64
```

From the provided list, identify columns with missing values comprising less than 13%:

AMT_ANNUITY AMT_GOODS_PRICE NAME_TYPE_SUITE CNT_FAM_MEMBERS EXT_SOURCE_2 OBS_30_CNT_SOCIAL_CIRCLE DEF_30_CNT_SOCIAL_CIRCLE OBS_60_CNT_SOCIAL_CIRCLE DEF_60_CNT_SOCIAL_CIRCLE DAYS_LAST_PHONE_CHANGE Now, let's analyze each field individually.

AMT_ANNUITY

```
# AMT_ANNUITY
print(application_data.AMT_ANNUITY.head()) # correct datatype
print(application_data.AMT_ANNUITY.describe())
application_data.boxplot(column=['AMT_ANNUITY'])
plt.show()
# from box plot it seems, it has lot of outliers so considering median
measure
application_data.AMT_ANNUITY.median()
# we can impute 24903(median) value in place of missing values

0    24700.5
1    35698.5
2     6750.0
3    29686.5
4    21865.5
Name: AMT_ANNUITY, dtype: float64
count    307499.000000
mean      27108.573909
std       14493.737315
min        1615.500000
25%       16524.000000
50%       24903.000000
75%       34596.000000
max      258025.500000
Name: AMT_ANNUITY, dtype: float64
```

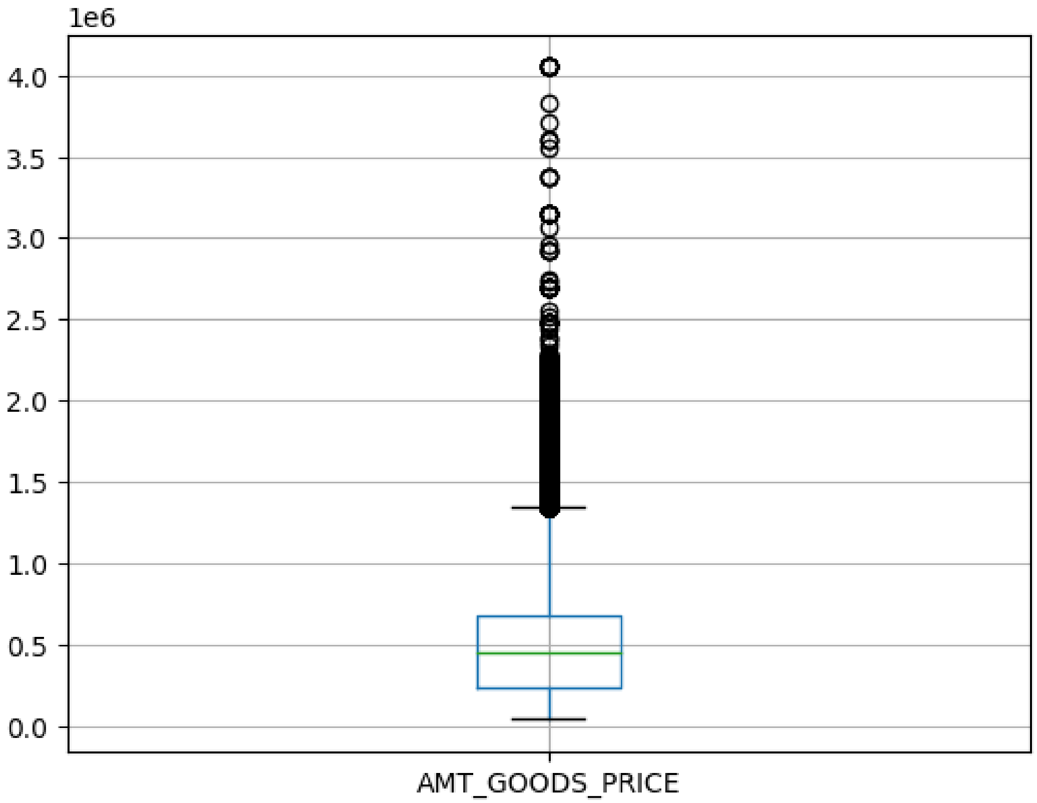


```
24903.0
```

AMT_GOODS_PRICE

```
# AMT_GOODS_PRICE
print(application_data.AMT_GOODS_PRICE.head()) # correct datatype
print(application_data.AMT_GOODS_PRICE.describe())
application_data.boxplot(column=['AMT_GOODS_PRICE'])
plt.show()
# from box plot it seems, it has lot of outliers so considering median
measure
application_data.AMT_GOODS_PRICE.median()
# we can impute 450000.0 value in place of missing values

0     351000.0
1    1129500.0
2     135000.0
3     297000.0
4     513000.0
Name: AMT_GOODS_PRICE, dtype: float64
count    3.072330e+05
mean     5.383962e+05
std      3.694465e+05
min      4.050000e+04
25%      2.385000e+05
```

```
50%      4.500000e+05
75%      6.795000e+05
max      4.050000e+06
Name: AMT_GOODS_PRICE, dtype: float64
```



```
450000.0
```

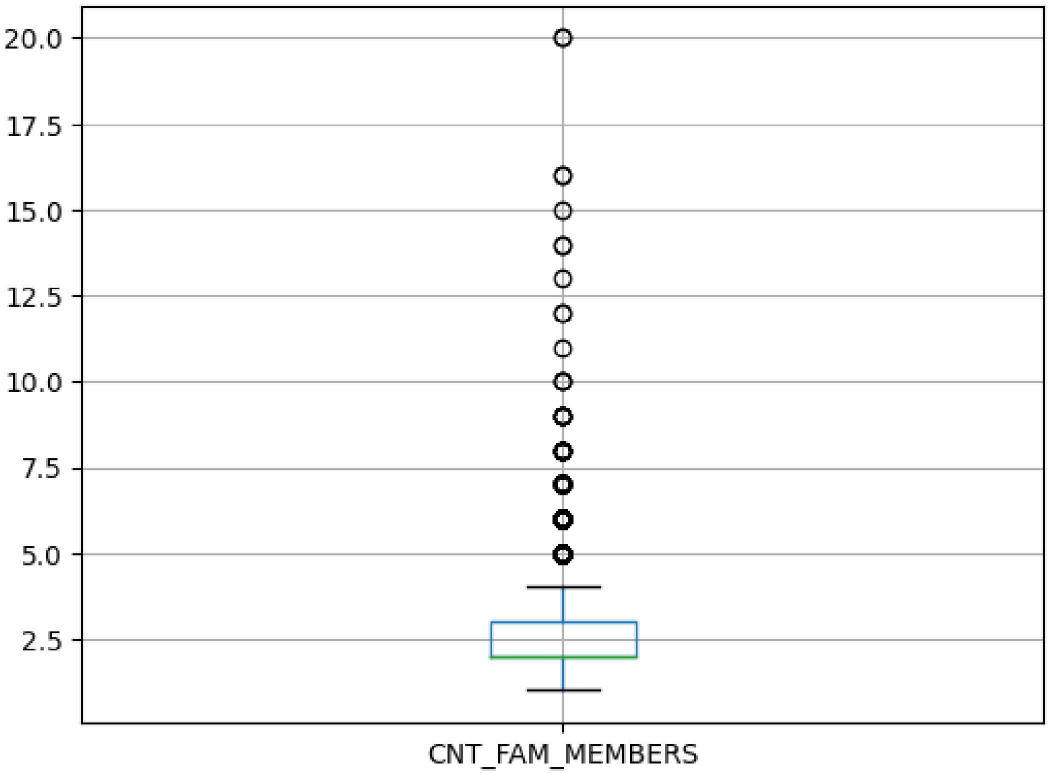
NAME_TYPE_SUITE

```
# NAME_TYPE_SUITE
print(application_data.NAME_TYPE_SUITE.head()) # correct datatype
print(application_data.NAME_TYPE_SUITE.describe())
# since it is acategorical value, considering mode measure to impute
missing values
print(application_data.NAME_TYPE_SUITE.mode())
# considering the value to be imputed is - Unaccompanied

0    Unaccompanied
1           Family
2    Unaccompanied
3    Unaccompanied
4    Unaccompanied
Name: NAME_TYPE_SUITE, dtype: object
count            306219
```

```
unique                    7
top       Unaccompanied
freq             248526
Name: NAME_TYPE_SUITE, dtype: object
0    Unaccompanied
Name: NAME_TYPE_SUITE, dtype: object
```

CNT_FAM_MEMBERS

```
#CNT_FAM_MEMBERS
print(application_data.CNT_FAM_MEMBERS.head()) # correct datatype
print(application_data.CNT_FAM_MEMBERS.describe())
application_data.boxplot(column=['CNT_FAM_MEMBERS'])
plt.show()
# from box plot it seems, it has lot of outliers so considering median
measure
application_data.CNT_FAM_MEMBERS.median()
# we can impute "2.0" value in place of missing values

0    1.0
1    2.0
2    1.0
3    2.0
4    1.0
Name: CNT_FAM_MEMBERS, dtype: float64
count    307509.000000
mean          2.152665
std           0.910682
min           1.000000
25%           2.000000
50%           2.000000
75%           3.000000
max          20.000000
Name: CNT_FAM_MEMBERS, dtype: float64
```

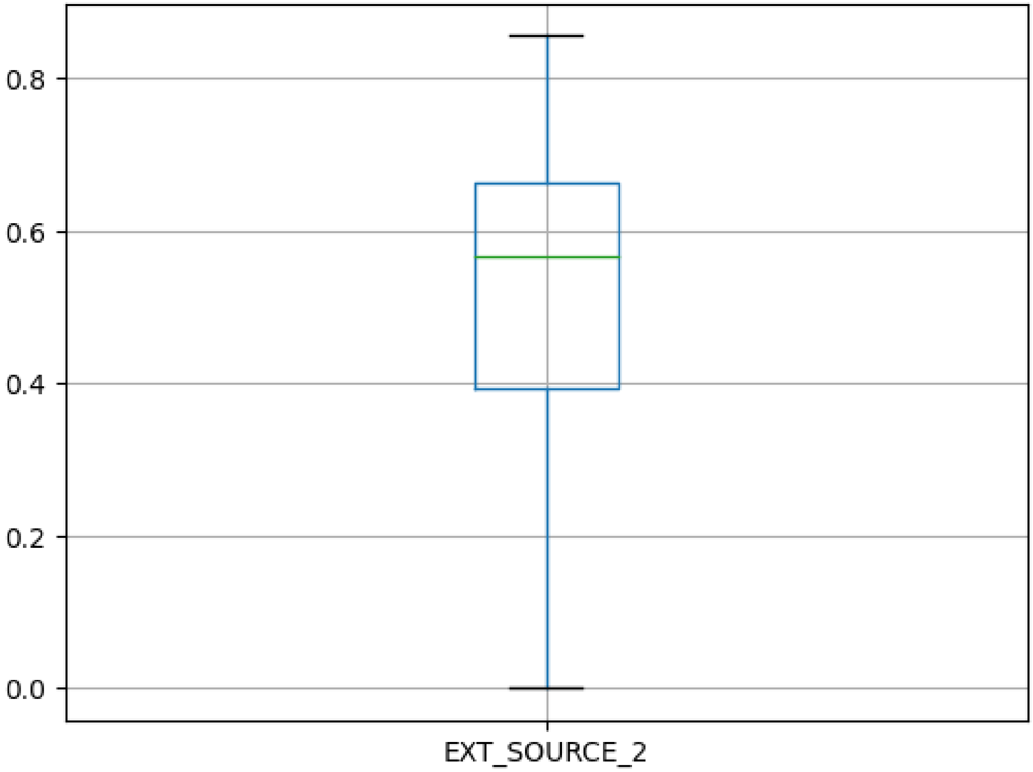


```
2.0
```

EXT_SOURCE_2

```
#EXT_SOURCE_2
print(application_data.EXT_SOURCE_2.head()) # correct datatype
print(application_data.EXT_SOURCE_2.describe())
application_data.boxplot(column=['EXT_SOURCE_2'])
plt.show()
# from box plot it seems, mean and median are almost near and no outliers but there is some tilt towards outliers so go with median
application_data.EXT_SOURCE_2.median()
# so, we can impute 0.5659614260608526 value in place of missing values

0    0.262949
1    0.622246
2    0.555912
3    0.650442
4    0.322738
Name: EXT_SOURCE_2, dtype: float64
count    3.068510e+05
mean     5.143927e-01
std      1.910602e-01
min      8.173617e-08
```

```
25%      3.924574e-01
50%      5.659614e-01
75%      6.636171e-01
max      8.549997e-01
Name: EXT_SOURCE_2, dtype: float64
```



```
0.5659614260608526
```

```
# checking the datatypes of all the columns and change the data type
like negative age and date
print(application_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 81 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   SK_ID_CURR                    307511 non-null  int64
 1   TARGET                        307511 non-null  int64
 2   NAME_CONTRACT_TYPE            307511 non-null  object
 3   CODE_GENDER                   307511 non-null  object
 4   FLAG_OWN_CAR                  307511 non-null  object
 5   FLAG_OWN_REALTY               307511 non-null  object
 6   CNT_CHILDREN                  307511 non-null  int64
 7   AMT_INCOME_TOTAL              307511 non-null  float64
 8   AMT_CREDIT                    307511 non-null  float64
 9   AMT_ANNUITY                   307499 non-null  float64
 10  AMT_GOODS_PRICE               307233 non-null  float64
 11  NAME_TYPE_SUITE               306219 non-null  object
 12  NAME_INCOME_TYPE              307511 non-null  object
 13  NAME_EDUCATION_TYPE           307511 non-null  object
 14  NAME_FAMILY_STATUS            307511 non-null  object
 15  NAME_HOUSING_TYPE             307511 non-null  object
 16  REGION_POPULATION_RELATIVE    307511 non-null  float64
 17  DAYS_BIRTH                    307511 non-null  int64
 18  DAYS_EMPLOYED                 307511 non-null  int64
 19  DAYS_REGISTRATION             307511 non-null  float64
 20  DAYS_ID_PUBLISH               307511 non-null  int64
 21  FLAG_MOBIL                    307511 non-null  int64
 22  FLAG_EMP_PHONE                307511 non-null  int64
 23  FLAG_WORK_PHONE               307511 non-null  int64
 24  FLAG_CONT_MOBILE              307511 non-null  int64
 25  FLAG_PHONE                    307511 non-null  int64
 26  FLAG_EMAIL                    307511 non-null  int64
 27  OCCUPATION_TYPE               211120 non-null  object
 28  CNT_FAM_MEMBERS               307509 non-null  float64
 29  REGION_RATING_CLIENT          307511 non-null  int64
 30  REGION_RATING_CLIENT_W_CITY   307511 non-null  int64
 31  WEEKDAY_APPR_PROCESS_START    307511 non-null  object
 32  HOUR_APPR_PROCESS_START       307511 non-null  int64
 33  REG_REGION_NOT_LIVE_REGION    307511 non-null  int64
 34  REG_REGION_NOT_WORK_REGION    307511 non-null  int64
 35  LIVE_REGION_NOT_WORK_REGION   307511 non-null  int64
 36  REG_CITY_NOT_LIVE_CITY        307511 non-null  int64
 37  REG_CITY_NOT_WORK_CITY        307511 non-null  int64
 38  LIVE_CITY_NOT_WORK_CITY       307511 non-null  int64
 39  ORGANIZATION_TYPE             307511 non-null  object
 40  EXT_SOURCE_2                  306851 non-null  float64
 41  EXT_SOURCE_3                  246546 non-null  float64
 42  YEARS_BEGINEXPLUATATION_AVG   157504 non-null  float64
 43  FLOORSMAX_AVG                 154491 non-null  float64
 44  YEARS_BEGINEXPLUATATION_MODE  157504 non-null  float64
 45  FLOORSMAX_MODE                154491 non-null  float64
 46  YEARS_BEGINEXPLUATATION_MEDI  157504 non-null  float64
 47  FLOORSMAX_MEDI                154491 non-null  float64
 48  TOTALAREA_MODE                159080 non-null  float64
 49  EMERGENCYSTATE_MODE           161756 non-null  object
 50  OBS_30_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 51  DEF_30_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 52  OBS_60_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 53  DEF_60_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 54  DAYS_LAST_PHONE_CHANGE        307510 non-null  float64
 55  FLAG_DOCUMENT_2               307511 non-null  int64
 56  FLAG_DOCUMENT_3               307511 non-null  int64
 57  FLAG_DOCUMENT_4               307511 non-null  int64
 58  FLAG_DOCUMENT_5               307511 non-null  int64
 59  FLAG_DOCUMENT_6               307511 non-null  int64
 60  FLAG_DOCUMENT_7               307511 non-null  int64
 61  FLAG_DOCUMENT_8               307511 non-null  int64
 62  FLAG_DOCUMENT_9               307511 non-null  int64
 63  FLAG_DOCUMENT_10              307511 non-null  int64
 64  FLAG_DOCUMENT_11              307511 non-null  int64
 65  FLAG_DOCUMENT_12              307511 non-null  int64
 66  FLAG_DOCUMENT_13              307511 non-null  int64
 67  FLAG_DOCUMENT_14              307511 non-null  int64
 68  FLAG_DOCUMENT_15              307511 non-null  int64
 69  FLAG_DOCUMENT_16              307511 non-null  int64
 70  FLAG_DOCUMENT_17              307511 non-null  int64
 71  FLAG_DOCUMENT_18              307511 non-null  int64
 72  FLAG_DOCUMENT_19              307511 non-null  int64
 73  FLAG_DOCUMENT_20              307511 non-null  int64
 74  FLAG_DOCUMENT_21              307511 non-null  int64
 75  AMT_REQ_CREDIT_BUREAU_HOUR    265992 non-null  float64
 76  AMT_REQ_CREDIT_BUREAU_DAY     265992 non-null  float64
 77  AMT_REQ_CREDIT_BUREAU_WEEK    265992 non-null  float64
 78  AMT_REQ_CREDIT_BUREAU_MON     265992 non-null  float64
 79  AMT_REQ_CREDIT_BUREAU_QRT     265992 non-null  float64
 80  AMT_REQ_CREDIT_BUREAU_YEAR    265992 non-null  float64
dtypes: float64(27), int64(41), object(13)
memory usage: 190.0+ MB
None
```

```
application_data.head()
```

{"type":"dataframe","variable_name":"application_data"}

```
# finding count of unique values in each column
print(application_data.nunique().sort_values())
```

```
FLAG_DOCUMENT_3                      2
FLAG_PHONE                           2
FLAG_DOCUMENT_4                      2
FLAG_DOCUMENT_2                      2
REG_REGION_NOT_LIVE_REGION           2
REG_REGION_NOT_WORK_REGION           2
LIVE_REGION_NOT_WORK_REGION          2
REG_CITY_NOT_LIVE_CITY               2
REG_CITY_NOT_WORK_CITY               2
LIVE_CITY_NOT_WORK_CITY              2
FLAG_DOCUMENT_14                     2
FLAG_DOCUMENT_13                     2
FLAG_DOCUMENT_12                     2
FLAG_DOCUMENT_11                     2
FLAG_DOCUMENT_10                     2
```

```
FLAG_DOCUMENT_9                      2
FLAG_DOCUMENT_8                      2
FLAG_DOCUMENT_7                      2
EMERGENCYSTATE_MODE                  2
FLAG_DOCUMENT_6                      2
FLAG_CONT_MOBILE                     2
FLAG_WORK_PHONE                      2
FLAG_EMAIL                           2
FLAG_MOBIL                           2
TARGET                               2
NAME_CONTRACT_TYPE                   2
FLAG_OWN_CAR                         2
FLAG_OWN_REALTY                      2
FLAG_DOCUMENT_21                     2
FLAG_DOCUMENT_20                     2
FLAG_EMP_PHONE                       2
FLAG_DOCUMENT_19                     2
FLAG_DOCUMENT_5                      2
FLAG_DOCUMENT_18                     2
FLAG_DOCUMENT_15                     2
FLAG_DOCUMENT_16                     2
FLAG_DOCUMENT_17                     2
REGION_RATING_CLIENT_W_CITY          3
CODE_GENDER                          3
REGION_RATING_CLIENT                 3
NAME_EDUCATION_TYPE                  5
AMT_REQ_CREDIT_BUREAU_HOUR           5
NAME_HOUSING_TYPE                    6
NAME_FAMILY_STATUS                   6
WEEKDAY_APPR_PROCESS_START           7
NAME_TYPE_SUITE                      7
NAME_INCOME_TYPE                     8
DEF_60_CNT_SOCIAL_CIRCLE             9
AMT_REQ_CREDIT_BUREAU_WEEK           9
AMT_REQ_CREDIT_BUREAU_DAY            9
DEF_30_CNT_SOCIAL_CIRCLE            10
AMT_REQ_CREDIT_BUREAU_QRT           11
CNT_CHILDREN                        15
CNT_FAM_MEMBERS                     17
OCCUPATION_TYPE                     18
HOUR_APPR_PROCESS_START             24
AMT_REQ_CREDIT_BUREAU_MON           24
AMT_REQ_CREDIT_BUREAU_YEAR          25
FLOORSMAX_MODE                      25
OBS_60_CNT_SOCIAL_CIRCLE            33
OBS_30_CNT_SOCIAL_CIRCLE            33
FLOORSMAX_MEDI                      49
ORGANIZATION_TYPE                   58
REGION_POPULATION_RELATIVE          81
```

```
YEARS_BEGINEXPLUATATION_MODE       221
YEARS_BEGINEXPLUATATION_MEDI       245
YEARS_BEGINEXPLUATATION_AVG        285
FLOORSMAX_AVG                      403
EXT_SOURCE_3                       814
AMT_GOODS_PRICE                   1002
AMT_INCOME_TOTAL                  2548
DAYS_LAST_PHONE_CHANGE            3773
TOTALAREA_MODE                    5116
AMT_CREDIT                        5603
DAYS_ID_PUBLISH                   6168
DAYS_EMPLOYED                    12574
AMT_ANNUITY                      13672
DAYS_REGISTRATION                15688
DAYS_BIRTH                       17460
EXT_SOURCE_2                    119831
SK_ID_CURR                      307511
dtype: int64
```

```
# converting negative DAYS_BIRTH value to positive value
application_data['DAYS_BIRTH']=application_data['DAYS_BIRTH'].abs()
# converting negative DAYS_EMPLOYED value to positive value
application_data['DAYS_EMPLOYED']=application_data['DAYS_EMPLOYED'].abs()
# converting negative DAYS_REGISTRATION value to positive value
application_data['DAYS_REGISTRATION']=application_data['DAYS_REGISTRATION'].abs()
# converting negative DAYS_ID_PUBLISH value to positive value
application_data['DAYS_ID_PUBLISH']=application_data['DAYS_ID_PUBLISH'].abs()
# converting negative DAYS_LAST_PHONE_CHANGE value to positive value
application_data['DAYS_LAST_PHONE_CHANGE']=application_data['DAYS_LAST_PHONE_CHANGE'].abs()
application_data.head()
```

```
{"type":"dataframe","variable_name":"application_data"}
```

```
# conversion of columns integer to categorical
for col in application_data.columns:
    if application_data[col].nunique() <= 3: # here considering columns with 3 unique values as categorical variables
        application_data[col] = application_data[col].astype(object)

application_data.info()
application_data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 81 columns):
 #   Column                             Non-Null Count   Dtype
```

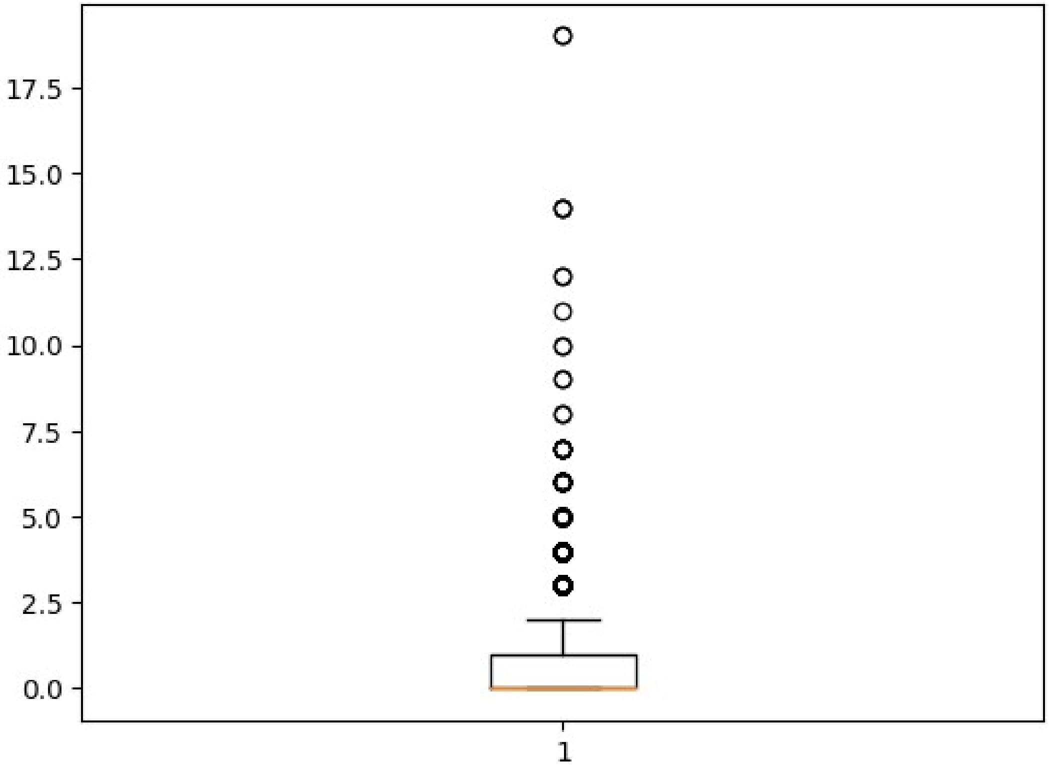
```
---  ------                        --------------   -----
 0   SK_ID_CURR                    307511 non-null  int64
 1   TARGET                        307511 non-null  object
 2   NAME_CONTRACT_TYPE            307511 non-null  object
 3   CODE_GENDER                   307511 non-null  object
 4   FLAG_OWN_CAR                  307511 non-null  object
 5   FLAG_OWN_REALTY               307511 non-null  object
 6   CNT_CHILDREN                  307511 non-null  int64
 7   AMT_INCOME_TOTAL              307511 non-null  float64
 8   AMT_CREDIT                    307511 non-null  float64
 9   AMT_ANNUITY                   307499 non-null  float64
 10  AMT_GOODS_PRICE               307233 non-null  float64
 11  NAME_TYPE_SUITE               306219 non-null  object
 12  NAME_INCOME_TYPE              307511 non-null  object
 13  NAME_EDUCATION_TYPE           307511 non-null  object
 14  NAME_FAMILY_STATUS            307511 non-null  object
 15  NAME_HOUSING_TYPE             307511 non-null  object
 16  REGION_POPULATION_RELATIVE    307511 non-null  float64
 17  DAYS_BIRTH                    307511 non-null  int64
 18  DAYS_EMPLOYED                 307511 non-null  int64
 19  DAYS_REGISTRATION             307511 non-null  float64
 20  DAYS_ID_PUBLISH               307511 non-null  int64
 21  FLAG_MOBIL                    307511 non-null  object
 22  FLAG_EMP_PHONE                307511 non-null  object
 23  FLAG_WORK_PHONE               307511 non-null  object
 24  FLAG_CONT_MOBILE              307511 non-null  object
 25  FLAG_PHONE                    307511 non-null  object
 26  FLAG_EMAIL                    307511 non-null  object
 27  OCCUPATION_TYPE               211120 non-null  object
 28  CNT_FAM_MEMBERS               307509 non-null  float64
 29  REGION_RATING_CLIENT          307511 non-null  object
 30  REGION_RATING_CLIENT_W_CITY   307511 non-null  object
 31  WEEKDAY_APPR_PROCESS_START    307511 non-null  object
 32  HOUR_APPR_PROCESS_START       307511 non-null  int64
 33  REG_REGION_NOT_LIVE_REGION    307511 non-null  object
 34  REG_REGION_NOT_WORK_REGION    307511 non-null  object
 35  LIVE_REGION_NOT_WORK_REGION   307511 non-null  object
 36  REG_CITY_NOT_LIVE_CITY        307511 non-null  object
 37  REG_CITY_NOT_WORK_CITY        307511 non-null  object
 38  LIVE_CITY_NOT_WORK_CITY       307511 non-null  object
 39  ORGANIZATION_TYPE             307511 non-null  object
 40  EXT_SOURCE_2                  306851 non-null  float64
 41  EXT_SOURCE_3                  246546 non-null  float64
 42  YEARS_BEGINEXPLUATATION_AVG   157504 non-null  float64
 43  FLOORSMAX_AVG                 154491 non-null  float64
 44  YEARS_BEGINEXPLUATATION_MODE  157504 non-null  float64
 45  FLOORSMAX_MODE                154491 non-null  float64
 46  YEARS_BEGINEXPLUATATION_MEDI  157504 non-null  float64
 47  FLOORSMAX_MEDI                154491 non-null  float64
 48  TOTALAREA_MODE                 159080 non-null  float64
 49  EMERGENCYSTATE_MODE            161756 non-null  object
 50  OBS_30_CNT_SOCIAL_CIRCLE       306490 non-null  float64
 51  DEF_30_CNT_SOCIAL_CIRCLE       306490 non-null  float64
 52  OBS_60_CNT_SOCIAL_CIRCLE       306490 non-null  float64
 53  DEF_60_CNT_SOCIAL_CIRCLE       306490 non-null  float64
 54  DAYS_LAST_PHONE_CHANGE         307510 non-null  float64
 55  FLAG_DOCUMENT_2                307511 non-null  object
 56  FLAG_DOCUMENT_3                307511 non-null  object
 57  FLAG_DOCUMENT_4                307511 non-null  object
 58  FLAG_DOCUMENT_5                307511 non-null  object
 59  FLAG_DOCUMENT_6                307511 non-null  object
 60  FLAG_DOCUMENT_7                307511 non-null  object
 61  FLAG_DOCUMENT_8                307511 non-null  object
 62  FLAG_DOCUMENT_9                307511 non-null  object
 63  FLAG_DOCUMENT_10               307511 non-null  object
 64  FLAG_DOCUMENT_11               307511 non-null  object
 65  FLAG_DOCUMENT_12               307511 non-null  object
 66  FLAG_DOCUMENT_13               307511 non-null  object
 67  FLAG_DOCUMENT_14               307511 non-null  object
 68  FLAG_DOCUMENT_15               307511 non-null  object
 69  FLAG_DOCUMENT_16               307511 non-null  object
 70  FLAG_DOCUMENT_17               307511 non-null  object
 71  FLAG_DOCUMENT_18               307511 non-null  object
 72  FLAG_DOCUMENT_19               307511 non-null  object
 73  FLAG_DOCUMENT_20               307511 non-null  object
 74  FLAG_DOCUMENT_21               307511 non-null  object
 75  AMT_REQ_CREDIT_BUREAU_HOUR     265992 non-null  float64
 76  AMT_REQ_CREDIT_BUREAU_DAY      265992 non-null  float64
 77  AMT_REQ_CREDIT_BUREAU_WEEK     265992 non-null  float64
 78  AMT_REQ_CREDIT_BUREAU_MON      265992 non-null  float64
 79  AMT_REQ_CREDIT_BUREAU_QRT      265992 non-null  float64
 80  AMT_REQ_CREDIT_BUREAU_YEAR     265992 non-null  float64
dtypes: float64(27), int64(6), object(48)
memory usage: 190.0+ MB
```

{"type":"dataframe","variable_name":"application_data"}

Inspect numerical columns for outliers and identify them for a minimum of five variables. Include additional observations and explanations.

```
plt.boxplot(application_data['CNT_CHILDREN'])
plt.show()
# From box plot, we can conclude that there exists values which are
above upper whisker(maximum) considered to be as outliers.
Q1 = application_data['CNT_CHILDREN'].quantile(0.25)
Q3 = application_data['CNT_CHILDREN'].quantile(0.75)
IQR = Q3 - Q1
lowerwhisker=(Q1 - 1.5 * IQR)
```

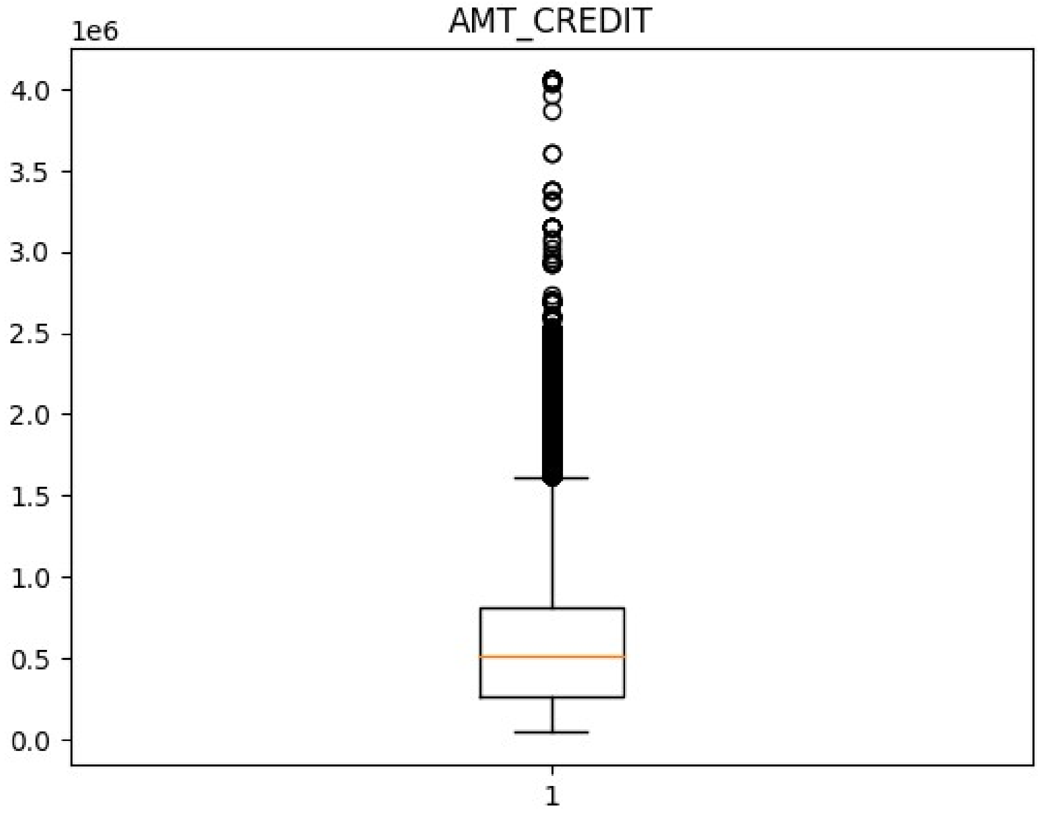
```
upperwhisker=(Q3 + 1.5 * IQR)
# According to Statictics the values above the upper whisker and below
the lower whisker are considered as outliers
#and as we can see in plot outliers are present only above the upper
wisker so considering them as outliers
print("The values greater than {} are considered to be outliers,since
count of children cannot be in decimals we can conclude that count
greater than 3 can be an outlier".format(upperwhisker))
```



```
The values greater than 2.5 are considered to be outliers,since count
of children cannot be in decimals we can conclude that count greater
than 3 can be an outlier
```

```
plt.boxplot(application_data['AMT_CREDIT'])
plt.title('AMT_CREDIT')
plt.show()
# From box plot, we can conclude that there exists values which are
above upper whisker(maximum) considered to be as outliers.
Q1 = application_data['AMT_CREDIT'].quantile(0.25)
Q3 = application_data['AMT_CREDIT'].quantile(0.75)
IQR = Q3 - Q1
lowerwhisker=(Q1 - 1.5 * IQR)
upperwhisker=(Q3 + 1.5 * IQR)

# the values above the upper whisker and below the lower whisker are
```

```
considered as outliers
#and as we can see in plot outliers are present only above the upper
wisker so considering them as outliers
#print("Lowerwhisker:{}".format(lowerwhisker))
'''according to statistics the the values less than lower whisker
value -537975.0 considered as outlier,
   as credit amount cannot be negative we consider amount greater than
1616625.0 as an outlier.'''
print("The amount credited greater than {} can be considered as an
outlier".format(upperwhisker))
```



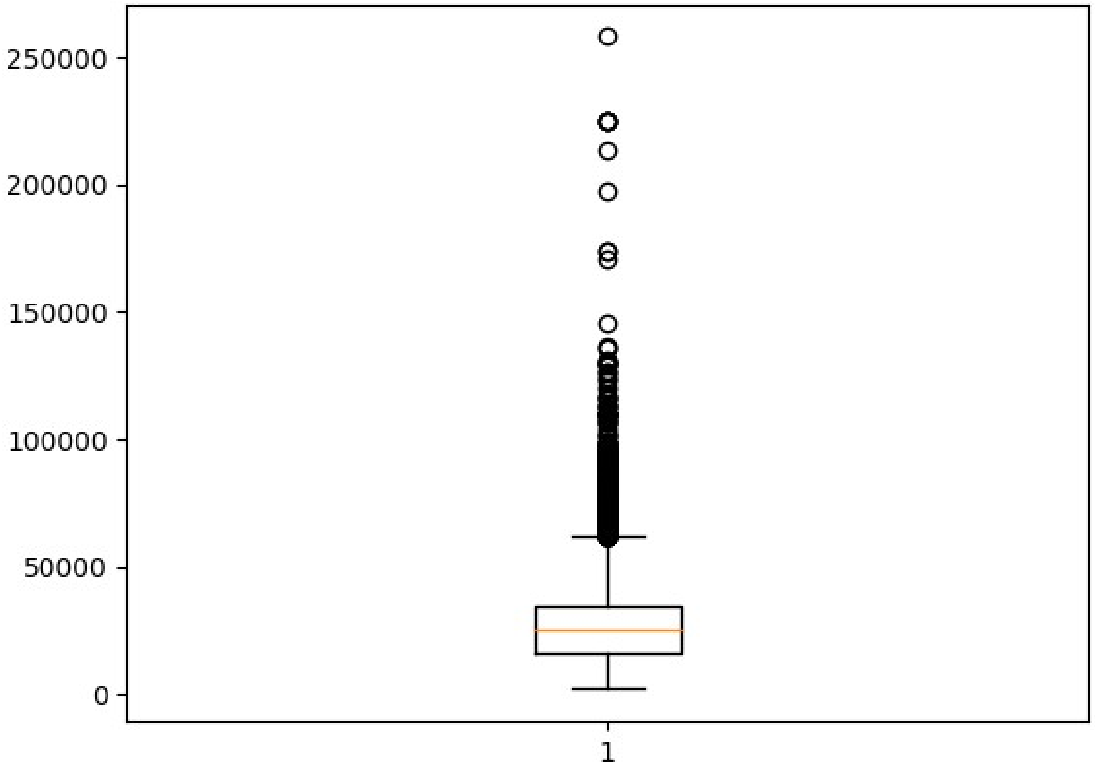
```
The amount credited greater than 1616625.0 can be considered as an
outlier
```

```
application_data['AMT_CREDIT'].describe()
application_data['AMT_CREDIT'].max()
```

```
4050000.0
```

```
data=application_data['AMT_ANNUITY']
filtered_data = data[~np.isnan(data)]
plt.boxplot(filtered_data)
plt.show()
# From box plot, we can conclude that there exists values which are
```

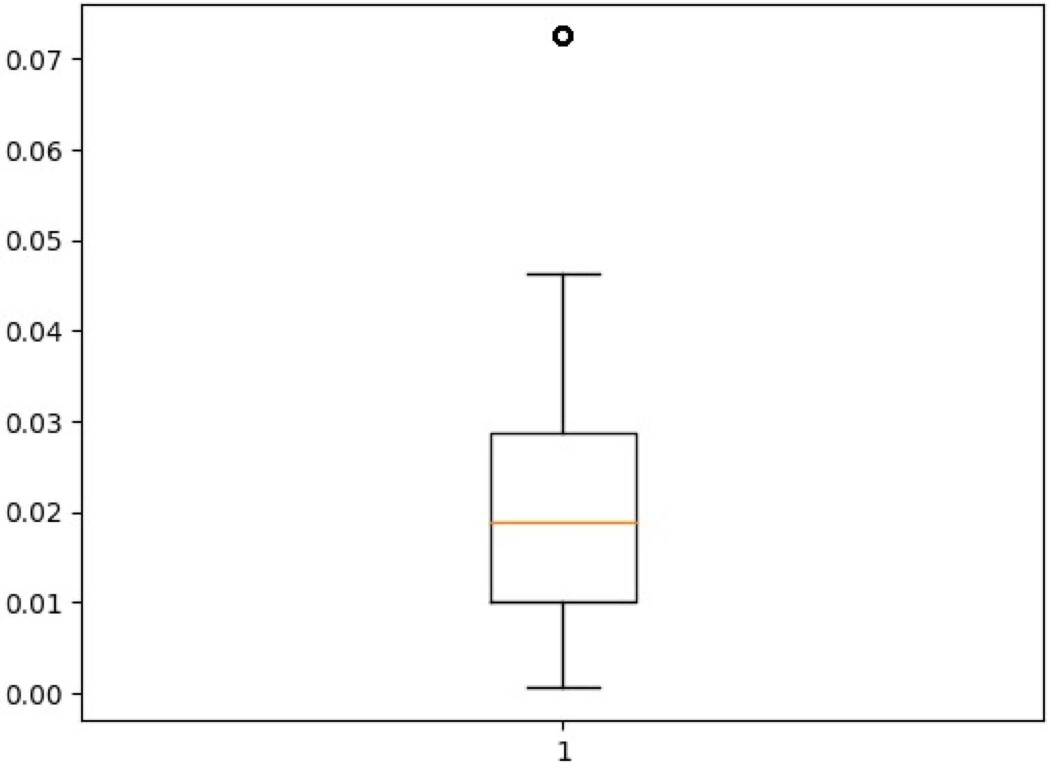
```
above upper whisker(maximum) considered to be as outliers.
Q1 = application_data['AMT_ANNUITY'].quantile(0.25)
Q3 = application_data['AMT_ANNUITY'].quantile(0.75)
IQR = Q3 - Q1
lowerwhisker=(Q1 - 1.5 * IQR)
upperwhisker=(Q3 + 1.5 * IQR)
# the values above the upper whisker and below the lower whisker are
considered as outliers
#and as we can see in plot outliers are present only above the upper
wisker so considering them as outliers
'''according to statistics the the values less than lower whisker
value -10584.0 considered as outlier,
   as amount cannot be negative we consider count greater than
61704.0 as an outlier.'''
print("Population relative count greater than {} is considered to be
an outlier".format(upperwhisker))
```



```
Population relative count greater than 61747.3125 is considered to be
an outlier
```

```
plt.boxplot(application_data['REGION_POPULATION_RELATIVE'])
plt.show()
# From box plot, we can conclude that there exists values which are
above upper whisker(maximum) considered to be as outliers.
Q1 = application_data['REGION_POPULATION_RELATIVE'].quantile(0.25)
```

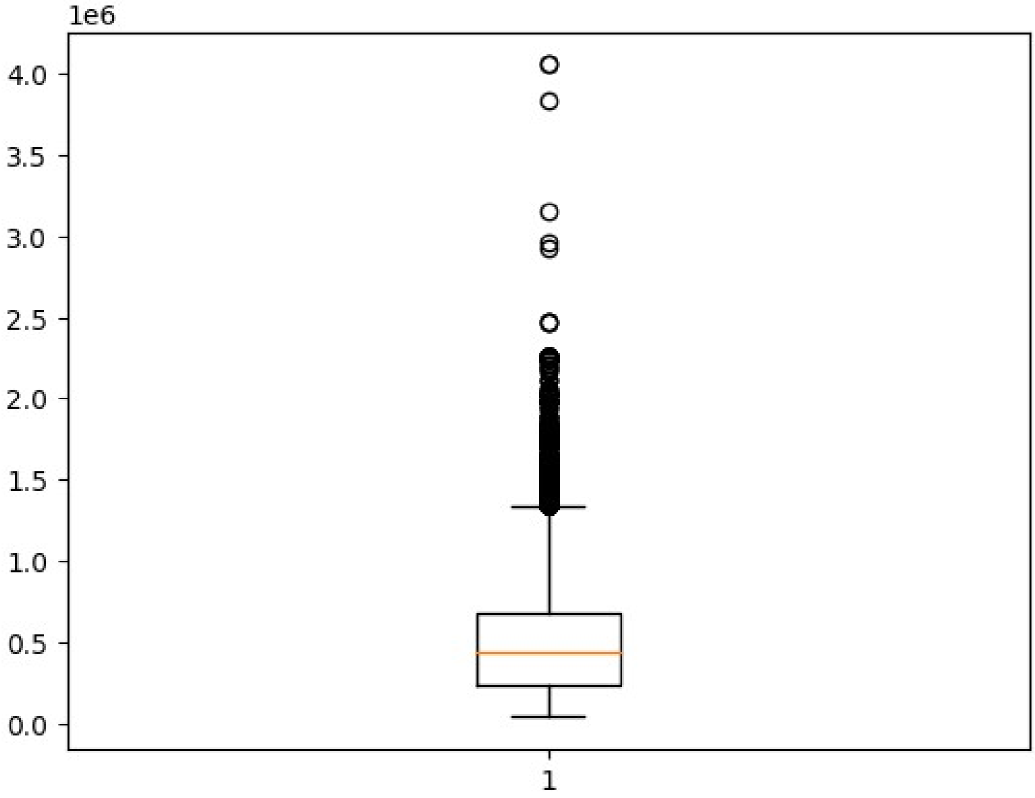
```
Q3 = application_data['REGION_POPULATION_RELATIVE'].quantile(0.75)
IQR = Q3 - Q1
lowerwhisker=(Q1 - 1.5 * IQR)
upperwhisker=(Q3 + 1.5 * IQR)
# the values above the upper whisker and below the lower whisker are
considered as outliers
#and as we can see in plot outliers are present only above the upper
wisker so considering them as outliers
'''according to statistics the the values less than lower whisker
value -0.017979500000000002 considered as outlier,
   as people relative cannot be negative we consider count greater
than  0.056648500000000004 as an outlier.'''
print("Population relative count greater than {} is considered to be
an outlier".format(upperwhisker))
```



```
Population relative count greater than 0.056648500000000004 is
considered to be an outlier
```

```
data=application_data['AMT_GOODS_PRICE']
filtered_data = data[~np.isnan(data)]
plt.boxplot(filtered_data)
plt.show()
# From box plot, we can conclude that there exists values which are
above upper whisker(maximum) considered to be as outliers.
Q1 = application_data['AMT_GOODS_PRICE'].quantile(0.25)
```

```
Q3 = application_data['AMT_GOODS_PRICE'].quantile(0.75)
IQR = Q3 - Q1
lowerwhisker=(Q1 - 1.5 * IQR)
upperwhisker=(Q3 + 1.5 * IQR)
# the values above the upper whisker and below the lower whisker are
considered as outliers
#and as we can see in plot outliers are present only above the upper
wisker so considering them as outliers
'''according to statistics the the values less than lower whisker
value -423000.0 considered as outlier,
   as amount cannot be negative we consider count greater than
1341000.0 as an outlier.'''
print("Population relative count greater than {} is considered to be
an outlier".format(upperwhisker))
```



```
Population relative count greater than 1341000.0 is considered to be
an outlier
```

```
application_data.head(10)
```

{"type":"dataframe","variable_name":"application_data"}

```
# Binning of continuous variables.Check if you need to bin any
variable in different categories.Do this for atleast 2 variables
```

```
# AMT_INCOME_TOTAL
q1=application_data['AMT_INCOME_TOTAL'].quantile(0.25)
q2=application_data['AMT_INCOME_TOTAL'].quantile(0.50)
q3=application_data['AMT_INCOME_TOTAL'].quantile(0.75)
m=application_data['AMT_INCOME_TOTAL'].max()

# Binning AMT_INCOME_TOTAL into AMT_INCOME_TOTAL_bin so we don't loose
data and have binned values
application_data['AMT_INCOME_TOTAL_bin'] =
pd.cut(application_data['AMT_INCOME_TOTAL'],[q1, q2, q3,m ], labels =
['Low', 'medium', 'High'])
print(application_data.AMT_INCOME_TOTAL_bin.value_counts())
```

```
AMT_INCOME_TOTAL_bin
medium    10870
High       9506
Low        7069
Name: count, dtype: int64
```

```
# AMT_CREDIT
q1=application_data['AMT_CREDIT'].quantile(0.25)
q2=application_data['AMT_CREDIT'].quantile(0.50)
q3=application_data['AMT_CREDIT'].quantile(0.75)
m=application_data['AMT_CREDIT'].max()

# Binning AMT_CREDIT into AMT_CREDIT_bin so we don't loose data and
have binned values
application_data['AMT_CREDIT_bin'] =
pd.cut(application_data['AMT_CREDIT'],[q1, q2, q3,m ], labels =
['Low', 'medium', 'High'])
print(application_data.AMT_CREDIT_bin.value_counts())
```

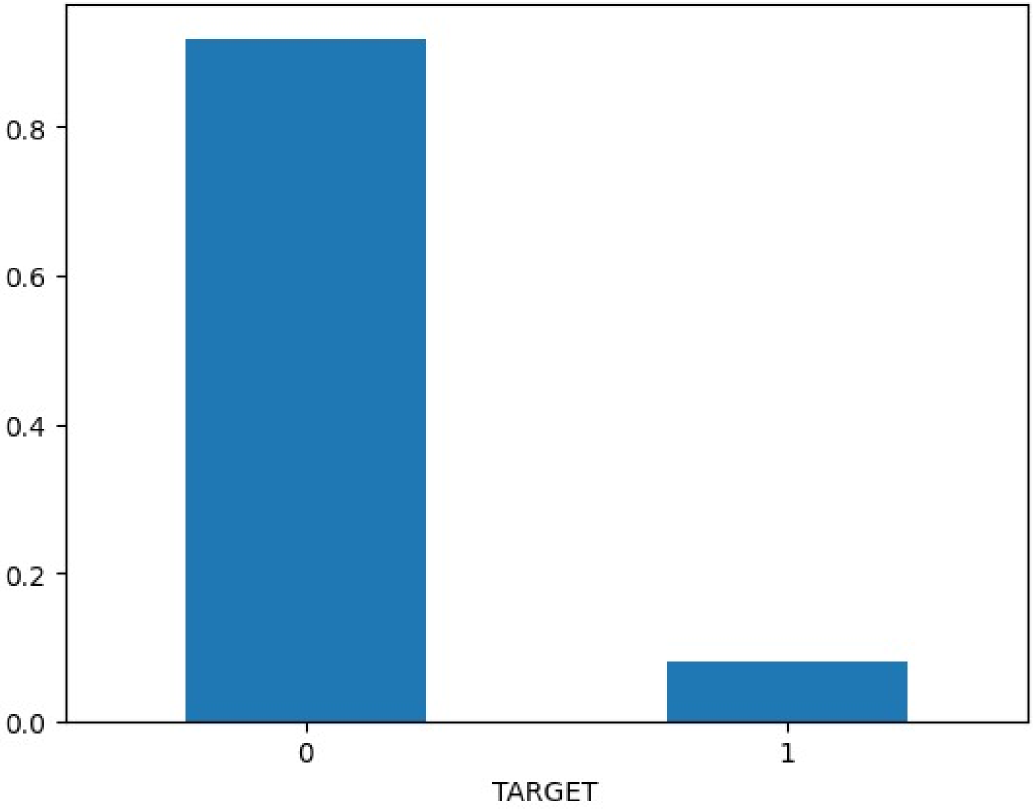
```
AMT_CREDIT_bin
medium    77786
High      75876
Low       75428
Name: count, dtype: int64
```

Analysis

```
application_data.head()
```

```
{"type":"dataframe","variable_name":"application_data"}
```

```
#Checking the imbalance percentage.
print(100*application_data.TARGET.value_counts()/
len(application_data))
(application_data.TARGET.value_counts()/
len(application_data)).plot.bar()
plt.xticks(rotation=0)
plt.show()
# In application_data there exists 91.927118% of "not default" and 
8.072882% of "default" customers.

TARGET
0    91.927118
1     8.072882
Name: count, dtype: float64
```



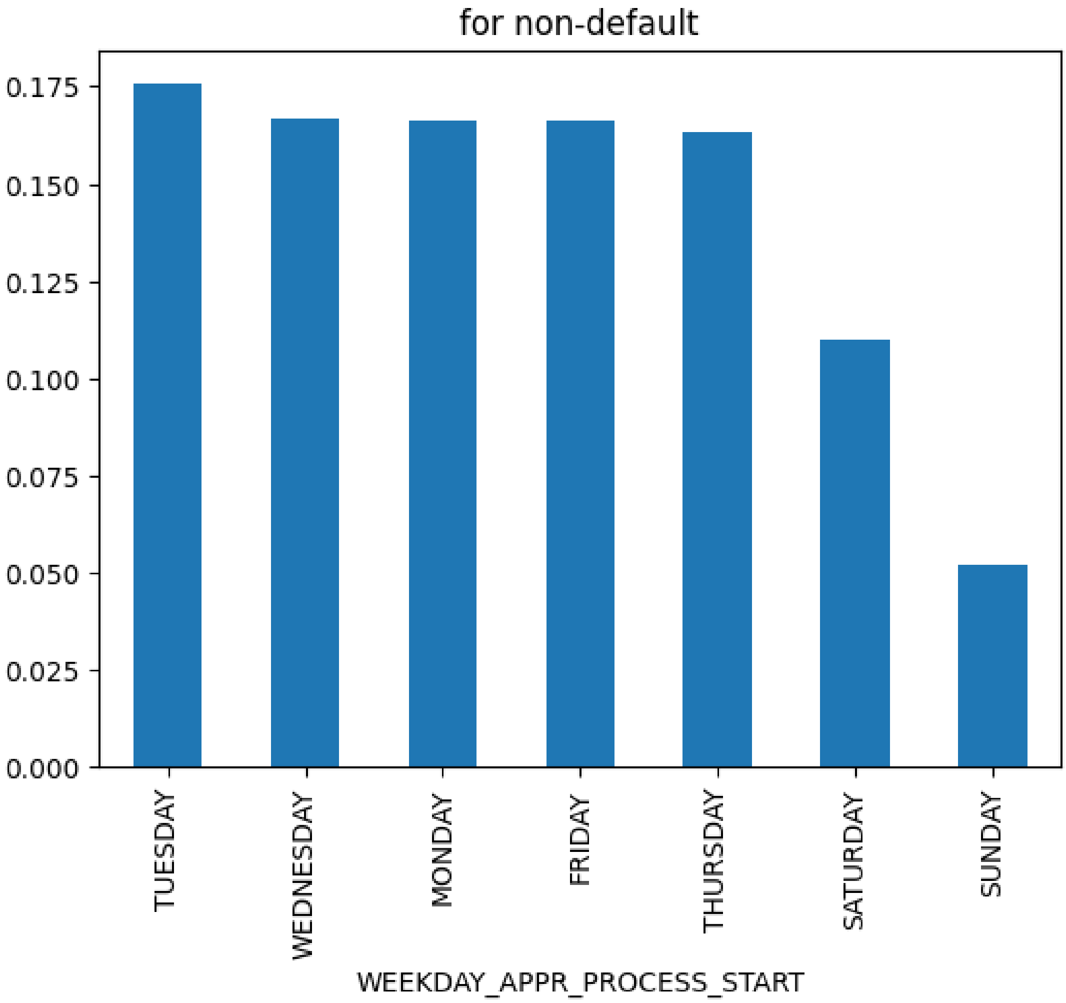
An unbalanced data set

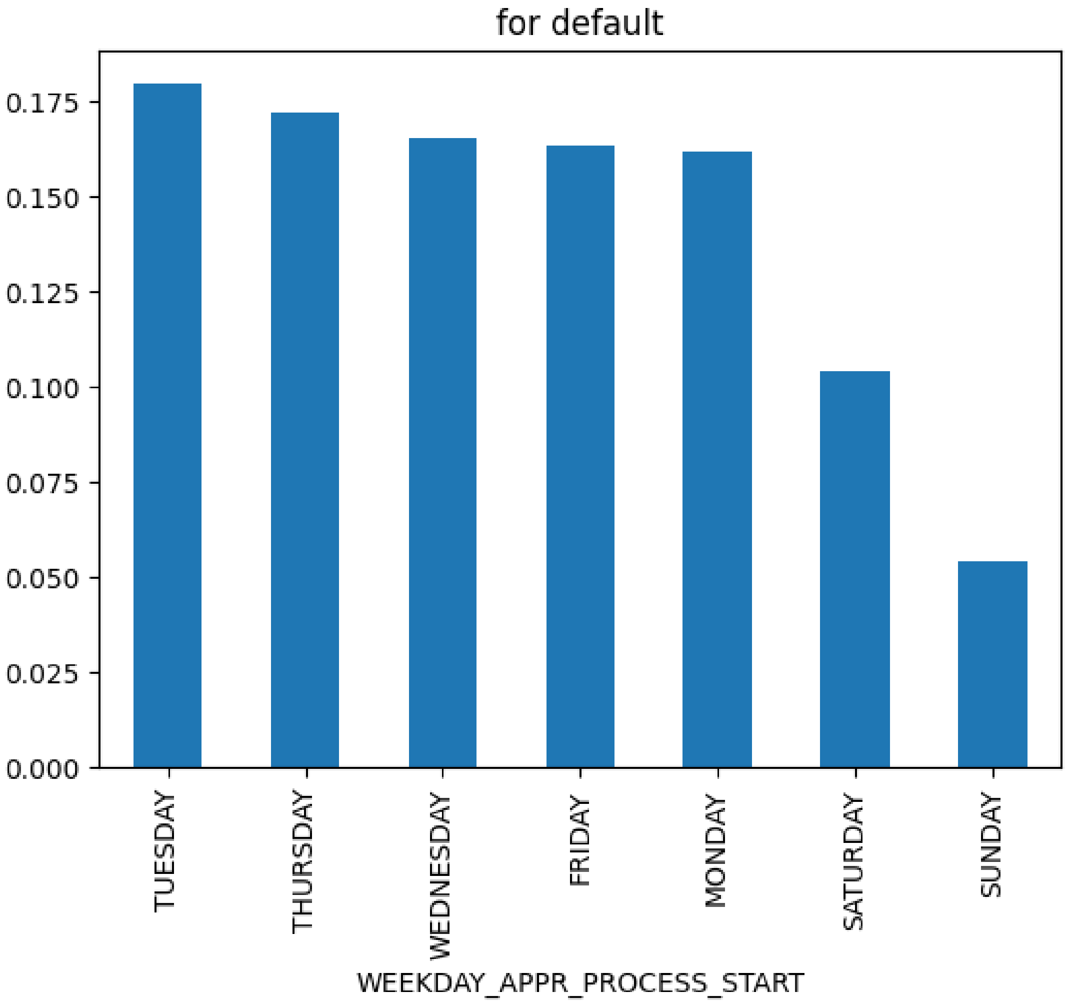
```
# Divide the data into two sets, i.e., Target-1 and Target-0
application_data_1 = application_data[application_data['TARGET']==1]
application_data_0 = application_data[application_data['TARGET']==0]
```

## Performing univariate analysis

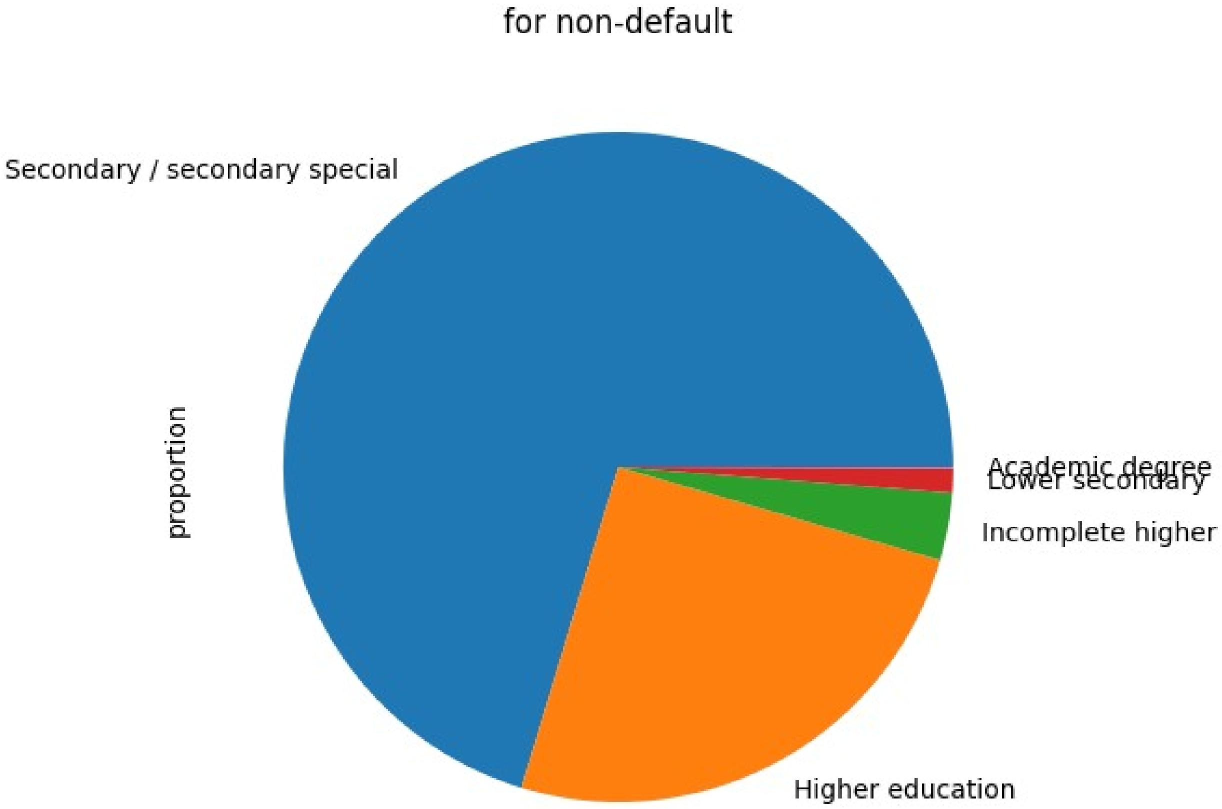
```
#Performing analysis for one column at a time
# perform univariate analysis for categoriacal variables for both 0 
and 1
# WEEKDAY_APPR_PROCESS_START (categorical ordered variable)
# for TARGET=0
application_data_0.WEEKDAY_APPR_PROCESS_START.value_counts(normalize=T
rue).plot.bar()
```

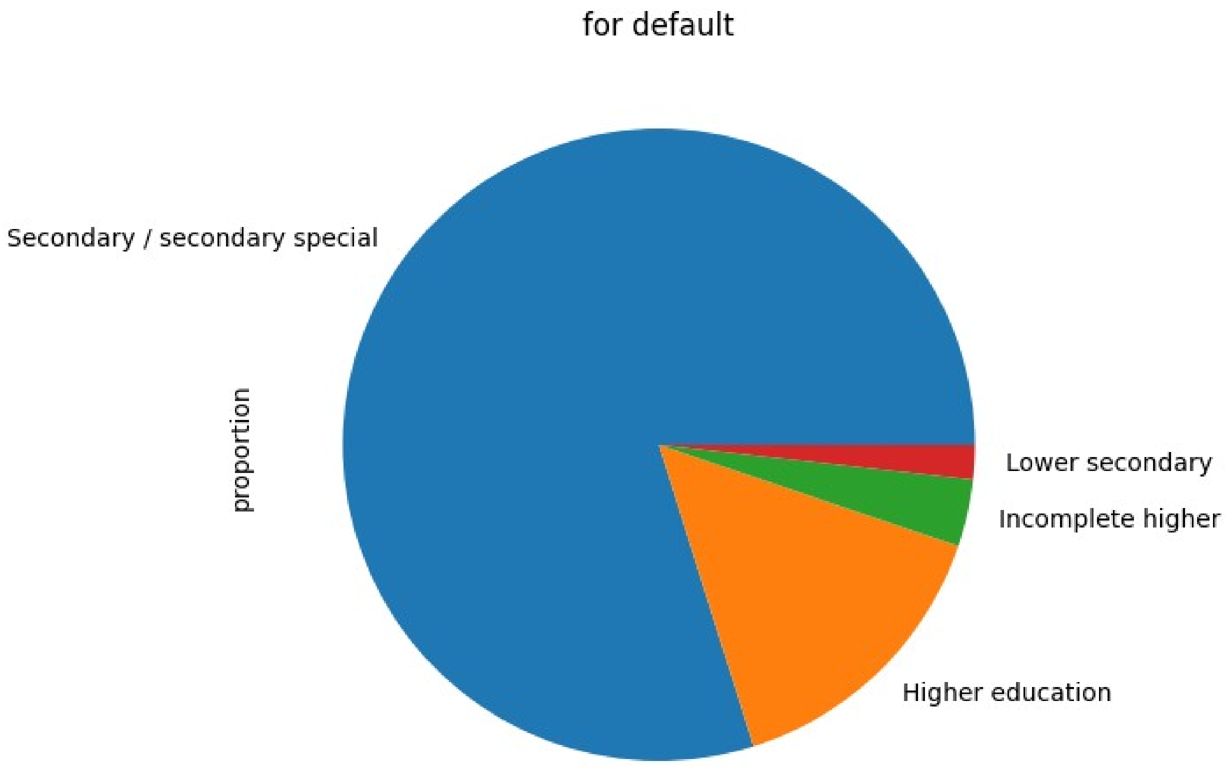
```
plt.title('for non-default')
plt.show()
# from the graph we can conclude that application starting processes
will be less in saturday and sunday.
# for TARGET=1
application_data_1.WEEKDAY_APPR_PROCESS_START.value_counts(normalize=T
rue).plot.bar()
plt.title('for default')
plt.show()
# from the graph we can conclude that application starting processes
are generally less in saturday and sunday.
```



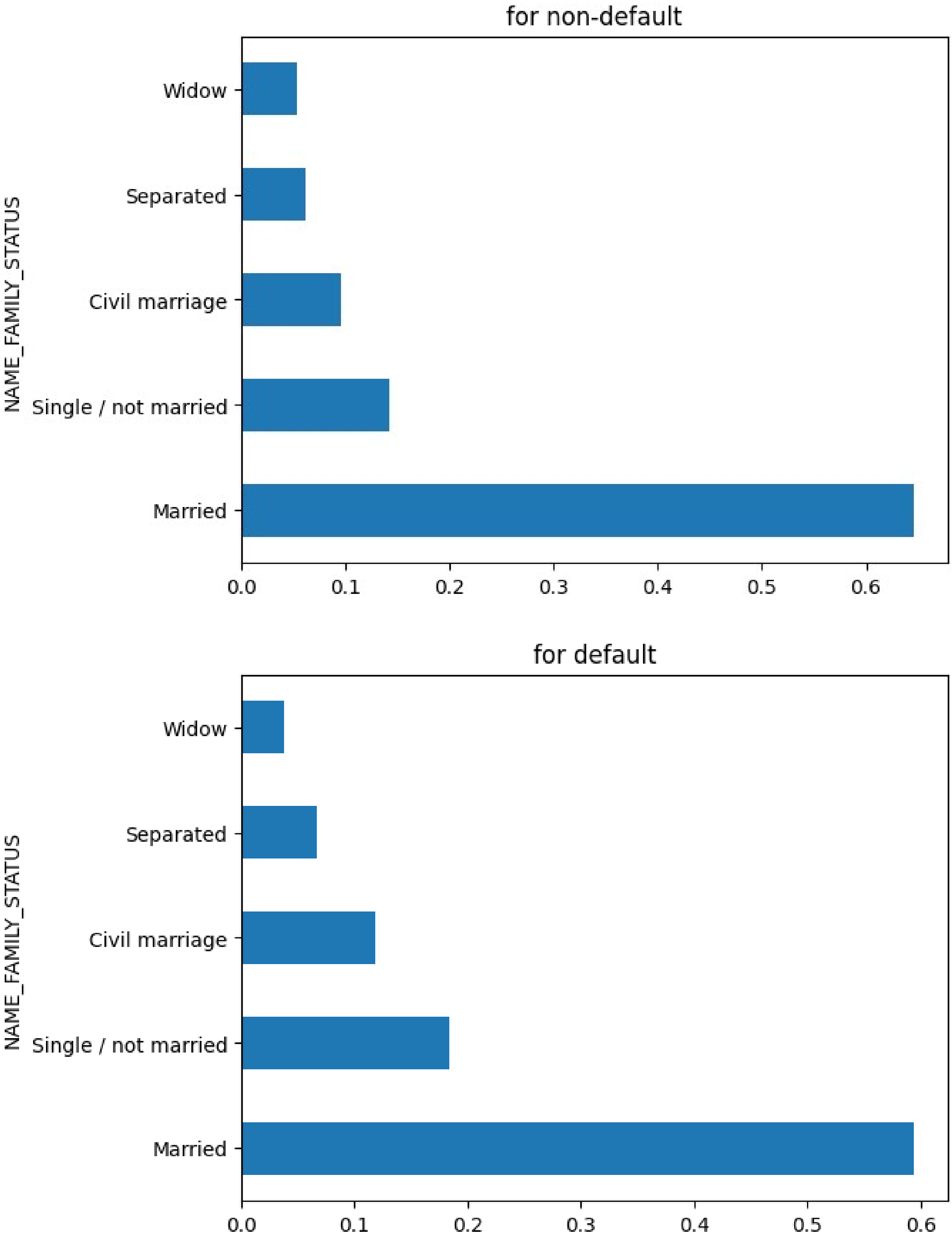


```
# NAME_EDUCATION_TYPE (categorical ordered variable)
# for Target=0
application_data_0.NAME_EDUCATION_TYPE.value_counts(normalize=True).pl
ot.pie()
plt.tight_layout()
plt.title('for non-default')
plt.show()
# from the plot below, we can conclude that secondary/special educated
people are applying loans in high in number.
# for Target=1
application_data_1.NAME_EDUCATION_TYPE.value_counts(normalize=True).pl
ot.pie()
plt.tight_layout()
plt.title('for default')
plt.show()
# from the plot below, we can conclude that secondary/special educated
people are applying loans high in number.
#and Academic degree educated people are applying loan in least count.
# for both target= 0 and 1
```

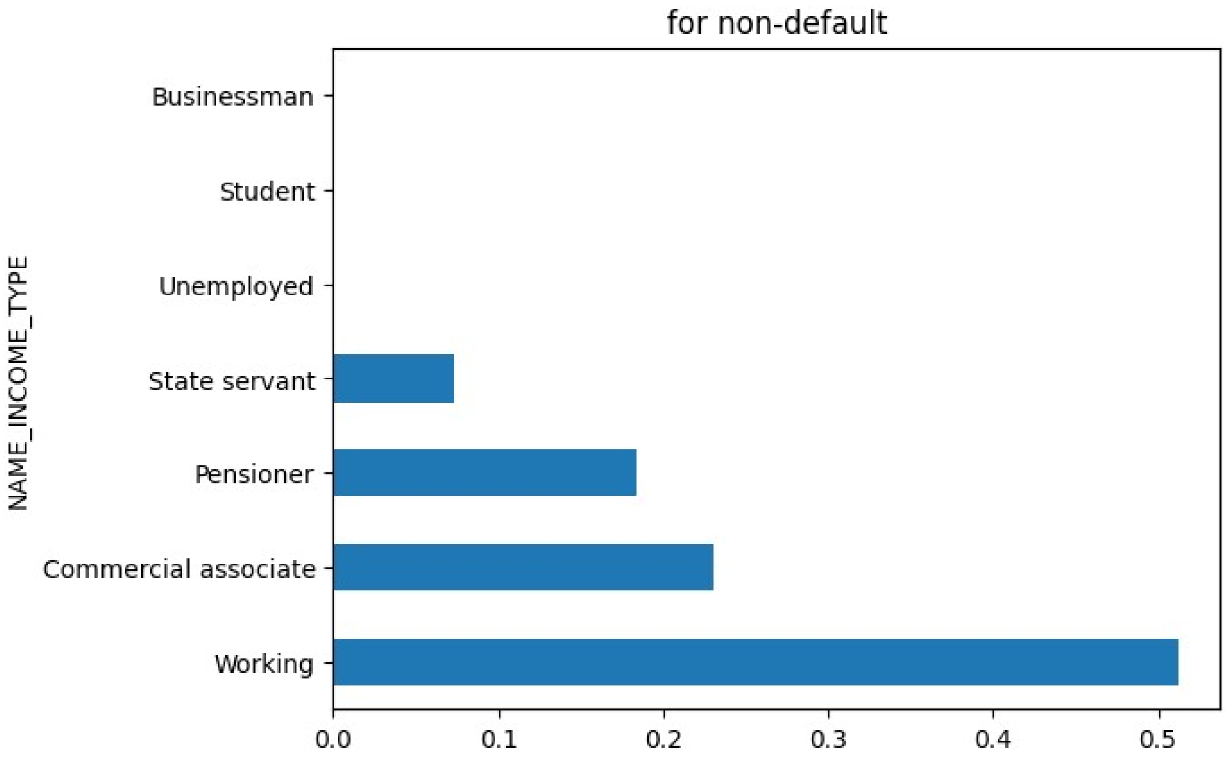

for non-default
Secondary / secondary special
proportion
Academic degree
Lower secondary
Incomplete higher
Higher education

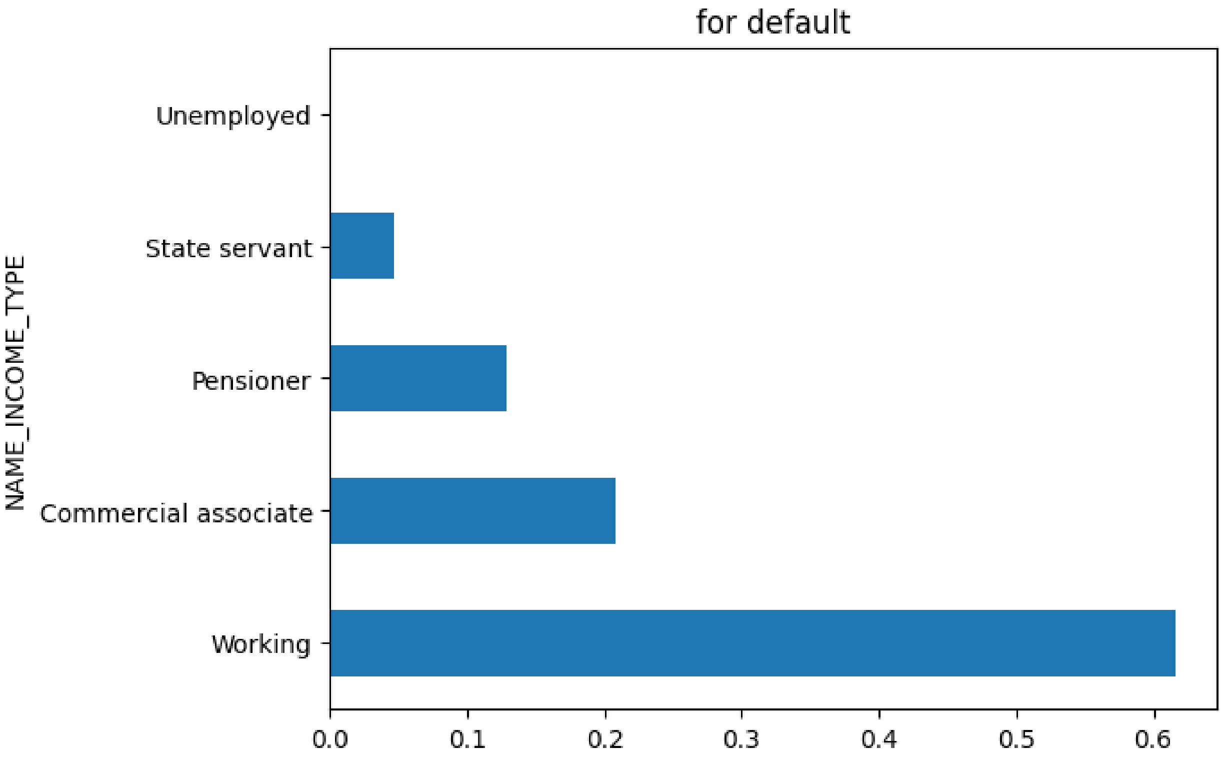


```
# NAME_FAMILY_STATUS
# for TARGET=0
application_data_0.NAME_FAMILY_STATUS.value_counts(normalize=True).plo
t.barh()
plt.title('for non-default')
plt.show()
# for TARGET=1
application_data_1.NAME_FAMILY_STATUS.value_counts(normalize=True).plo
t.barh()
plt.title('for default')
plt.show()
# the order of both default and not default customers is same i.e.,
Married,Single/not married,civil marriage,seperated,widow
# It also shows that there exists few(1 or 2) unknown values in not
default client family status.

# We can say more married people tend to take more Loan as compaired
to other categories
# and being married is not impacting default and not defaulting
```

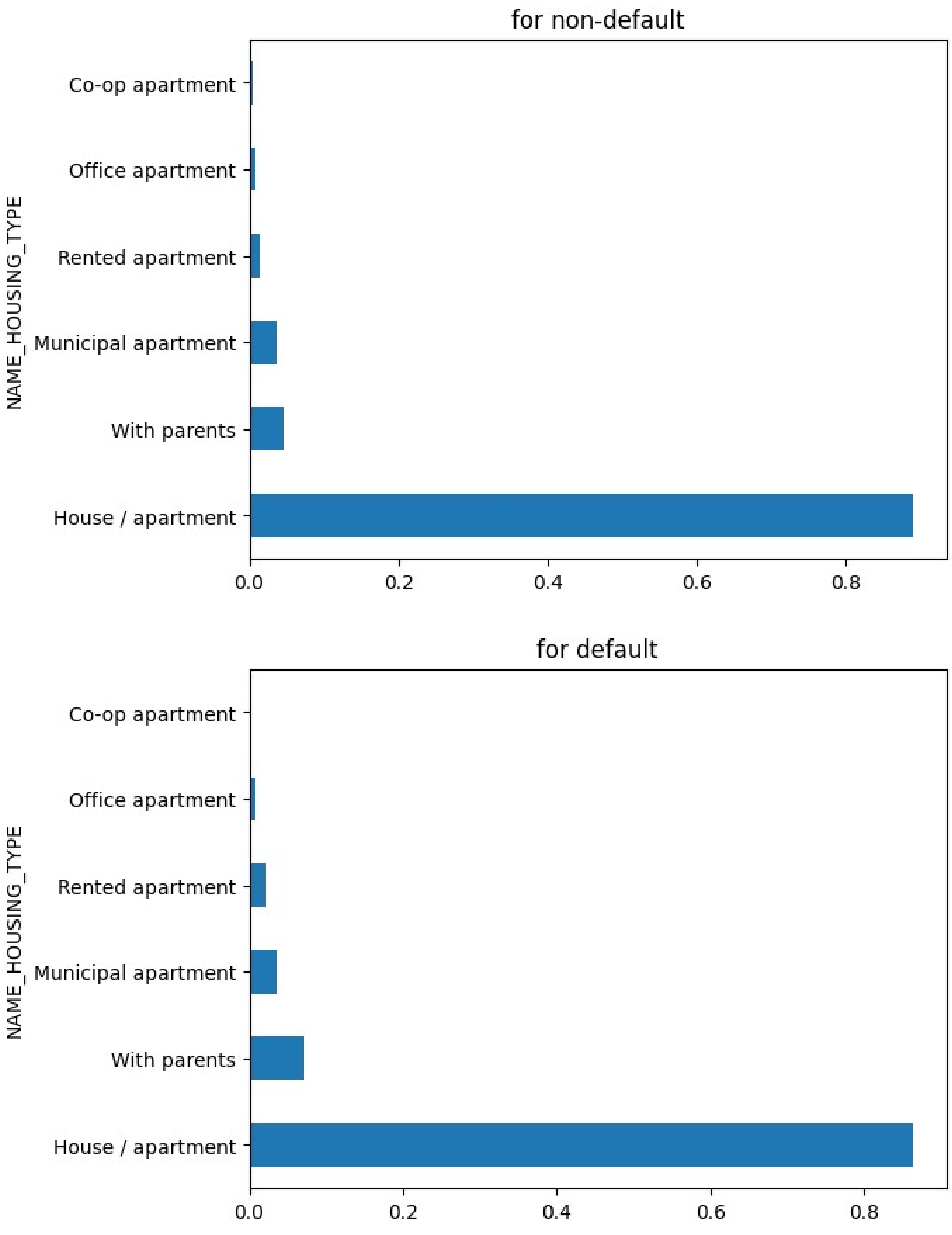

for non-default
NAME_FAMILY_STATUS
Widow
Separated
Civil marriage
Single / not married
Married
0.0
0.1
0.2
0.3
0.4
0.5
0.6
for default
NAME_FAMILY_STATUS
Widow
Separated
Civil marriage
Single / not married
Married
0.0
0.1
0.2
0.3
0.4
0.5
0.6

```
# NAME_INCOME_TYPE
# for TARGET=0
application_data_0.NAME_INCOME_TYPE.value_counts(normalize=True).plot.
barh()
plt.title('for non-default')
plt.show()
# for TARGET=1
application_data_1.NAME_INCOME_TYPE.value_counts(normalize=True).plot.
barh()
plt.title('for default')
plt.show()
# from the graphs below, we can conclude that
# Pensioner of not default case are high in number compared to
Pensioner of default case.
#It seems there exists both loss and profit due to Pension people to
the Bank.
# It also shows that majority of defaulters income type is working.
#and at the same time there is good income to bank from working
people.
```



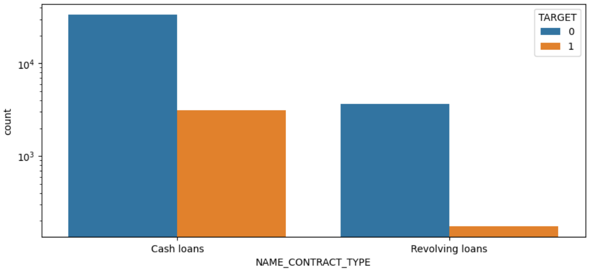


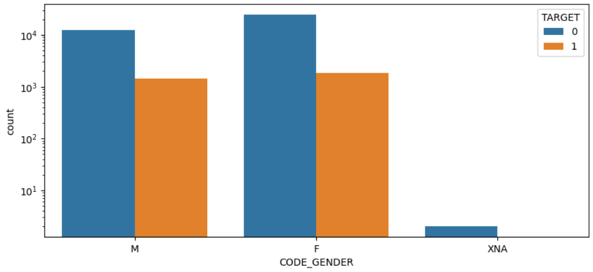
```
# NAME_HOUSING_TYPE
# for TARGET=0
application_data_0.NAME_HOUSING_TYPE.value_counts(normalize=True).plot
.barh()
plt.title('for non-default')
plt.show()
# for TARGET=1
application_data_1.NAME_HOUSING_TYPE.value_counts(normalize=True).plot
.barh()
plt.title('for default')
plt.show()
# from graph we can conclude that there exists people who have own
house
# lies in both default and non default.
```

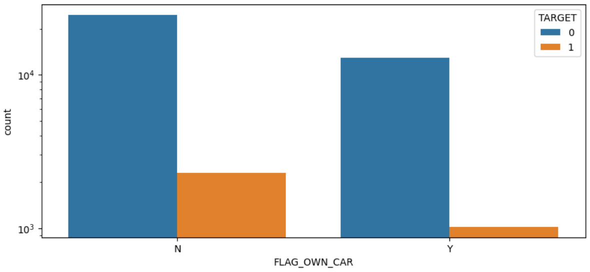


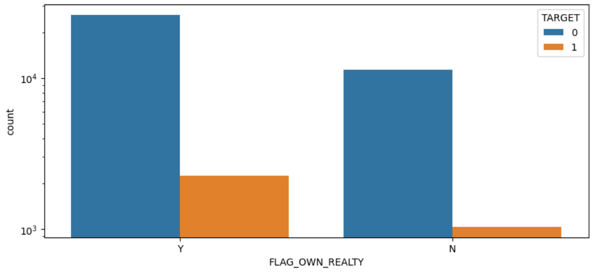
Compare the target variable across the categories of categorical variables against Target 0 and 1

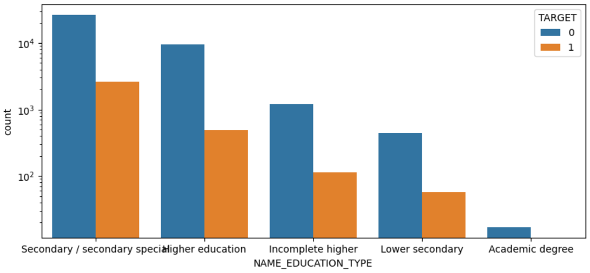
```
#considering 10 categorical columns
categorical_columns=['NAME_CONTRACT_TYPE','CODE_GENDER','FLAG_OWN_CAR'
,'FLAG_OWN_REALTY',

'NAME_EDUCATION_TYPE','NAME_FAMILY_STATUS','NAME_HOUSING_TYPE',

'WEEKDAY_APPR_PROCESS_START','AMT_CREDIT_bin','AMT_INCOME_TOTAL_bin']

plt.figure(figsize=(22,25))
for i in (enumerate(categorical_columns)):
    plt.subplot(len(categorical_columns)//2,2,i[0]+1)
    sns.countplot(x=i[1],hue='TARGET',data=application_data)
    plt.yscale('log')
    #plt.xticks(rotation=90)
plt.show()
#the XNA in Code_gender is not known if it is NA or a category so
leaving it as it is.
```

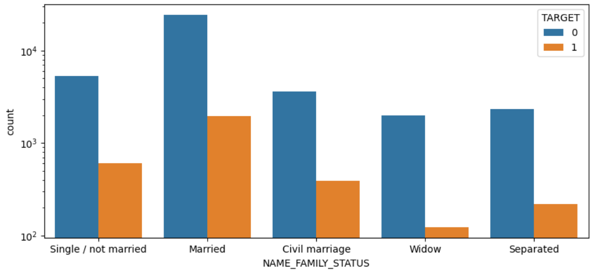


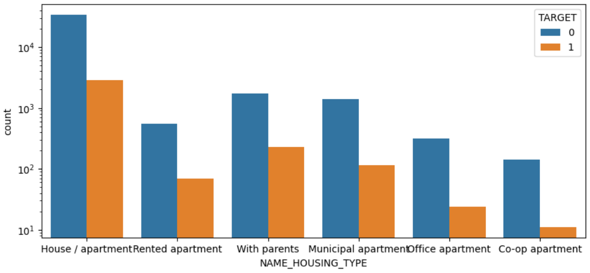


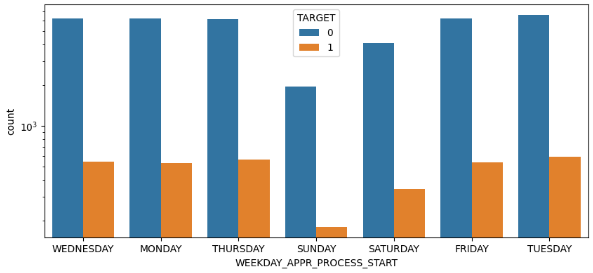


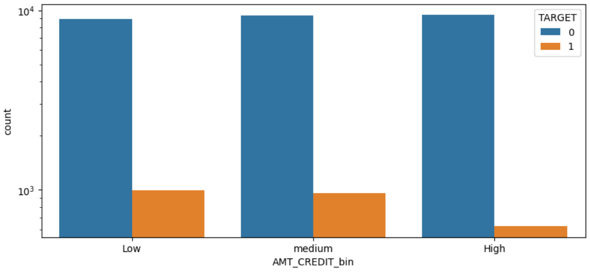


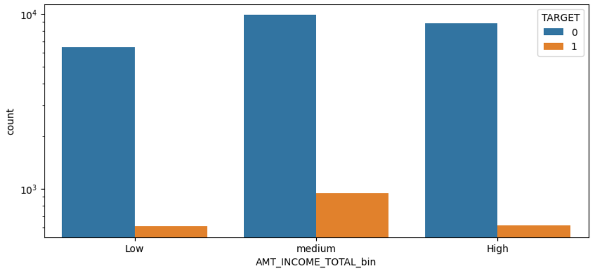












Conclusions/Insights

As we can see from graphs

- People with Medium total income are more likely to default
- People with high Credit amount are less likely to default
- People who started application process on sunday are less likely to default
- Saturday and sunday are less busy for bank in terms of loan applications

- People with house or appartment tend to take more loans
- We can say more married people tend to take more Loan as compaired to other categories
- we can conclude that secondary/special educated people are applying loans in high in number
- People with real estate tends to take more loans
- People who don't own a car tends to take more loans
- Female tends to take more loans
- People tend to take more cash loans, and default percentage of revolving loans is less

```
#considering 10 continous numerical columns
continous_columns=['AMT_ANNUITY','AMT_GOODS_PRICE','CNT_FAM_MEMBERS',

'DAYS_LAST_PHONE_CHANGE','DAYS_ID_PUBLISH','DAYS_BIRTH','HOUR_APPR_PRO
CESS_START',
                  'DAYS_EMPLOYED','AMT_CREDIT','AMT_INCOME_TOTAL']
plt.figure(figsize=(22,25))
for i in (enumerate(continous_columns)):
    plt.subplot(len(continous_columns)//2,2,i[0]+1)

sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')

sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
plt.show()
```

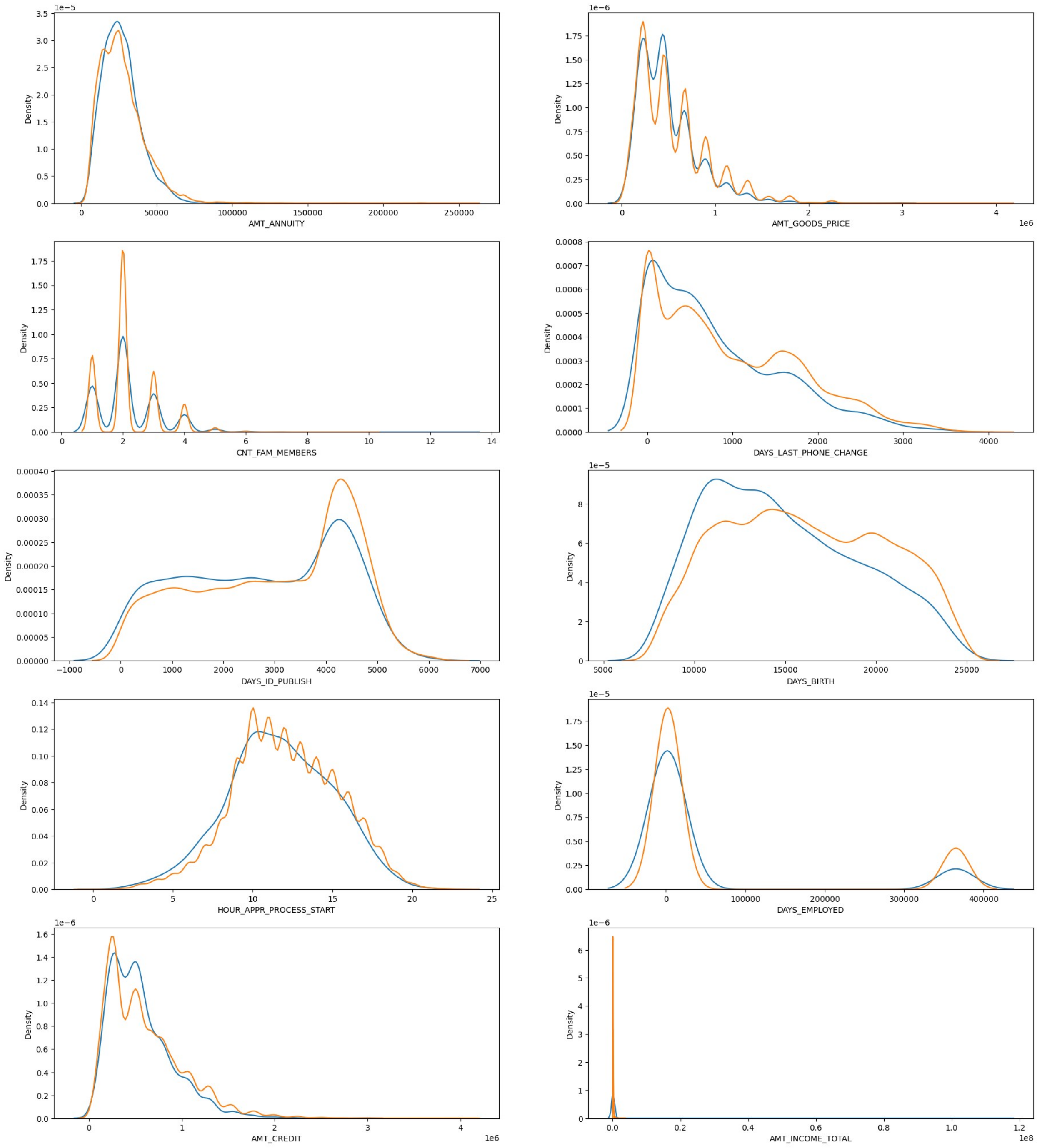
```
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).
```

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
```

```
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
```

```
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
<ipython-input-52-e376b429858d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_1[i[1]].dropna(),hist=False,label='Targe
t : default')
<ipython-input-52-e376b429858d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(application_data_0[i[1]].dropna(),hist=False,label='Targe
t : no default')
```



## Insights

As we can see from graphs

- People with lower total income are more likely to default
- People who just got employed tends to take more loans
- People who retired tends to take more loans
- High number of applications are filed in 10 AM to 2 PM

- People with age between 27yrs(10000-days) and 41(15000-days) yrs tend to take more loans
- People whose id(s) got published between 4000 days and 5000 days ago tend to take more loans
- nuclear family tends to take more loans
- for less goods amount people take loans
- low amount annuity has high number of loans

## Performing Bi-variate analysis

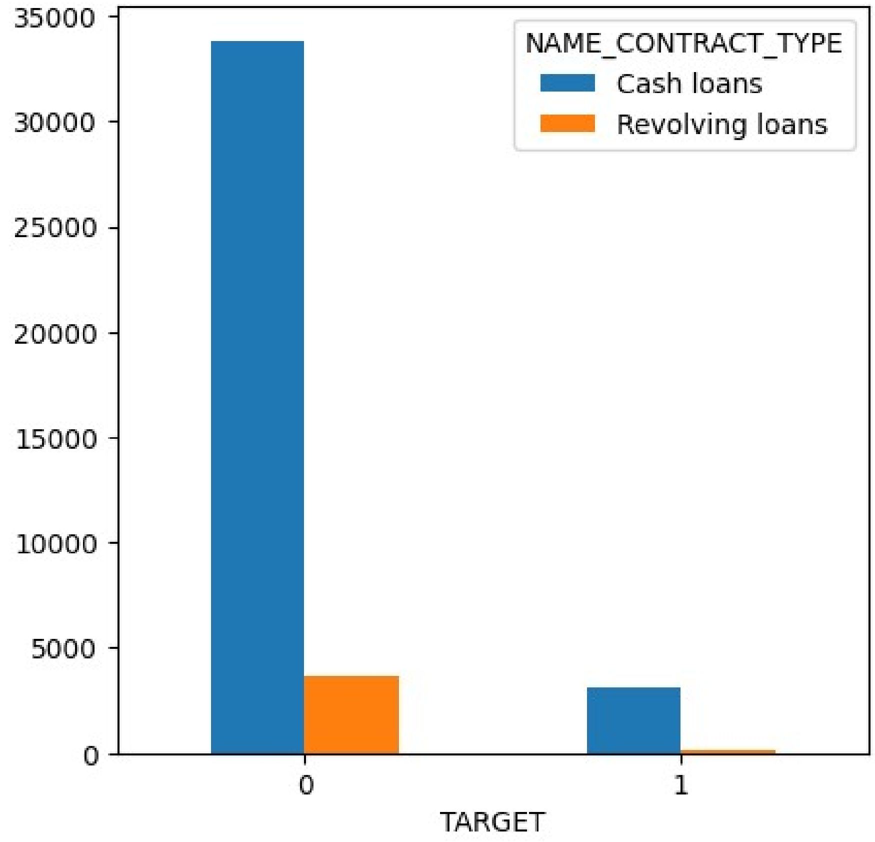
```
application_data.head()
```

```
{"type":"dataframe","variable_name":"application_data"}
```

Bi-variate categorical plots

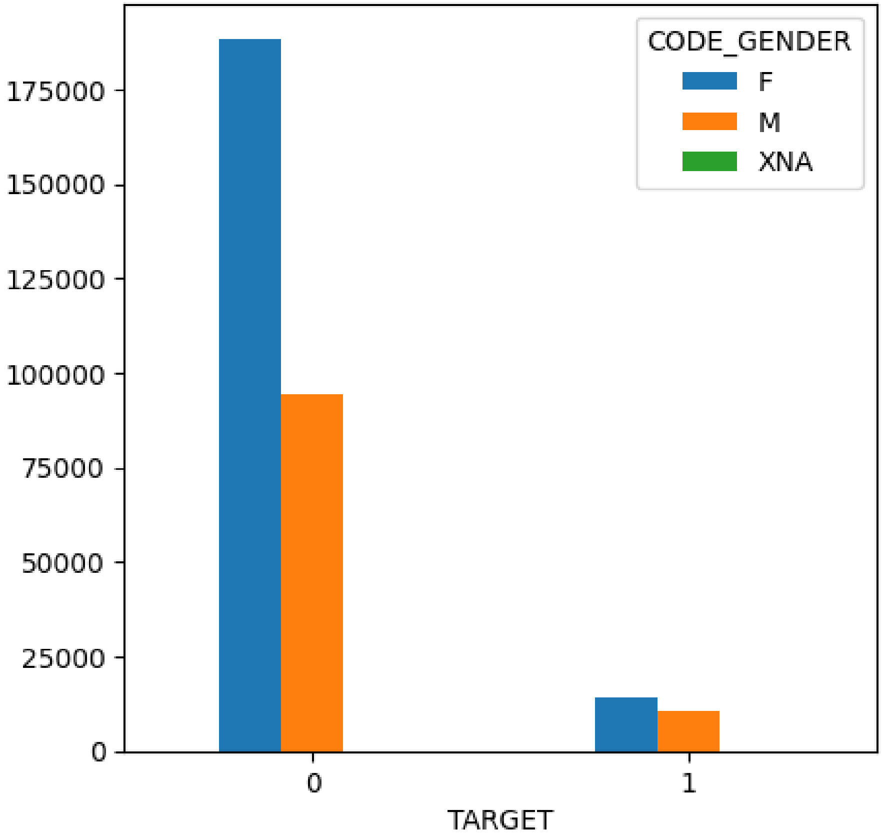
```
#Bi-variate categorical plots

table_1=
pd.crosstab(index=application_data['TARGET'],columns=application_data[
'NAME_CONTRACT_TYPE'])
print(table_1)
table_1.plot(kind="bar", figsize=(5,5),stacked=False)
plt.xticks(rotation=0)
plt.show()
# High number of cash loans
```

```
NAME_CONTRACT_TYPE  Cash loans  Revolving loans
TARGET
0                        33810             3673
1                         3125              174
```

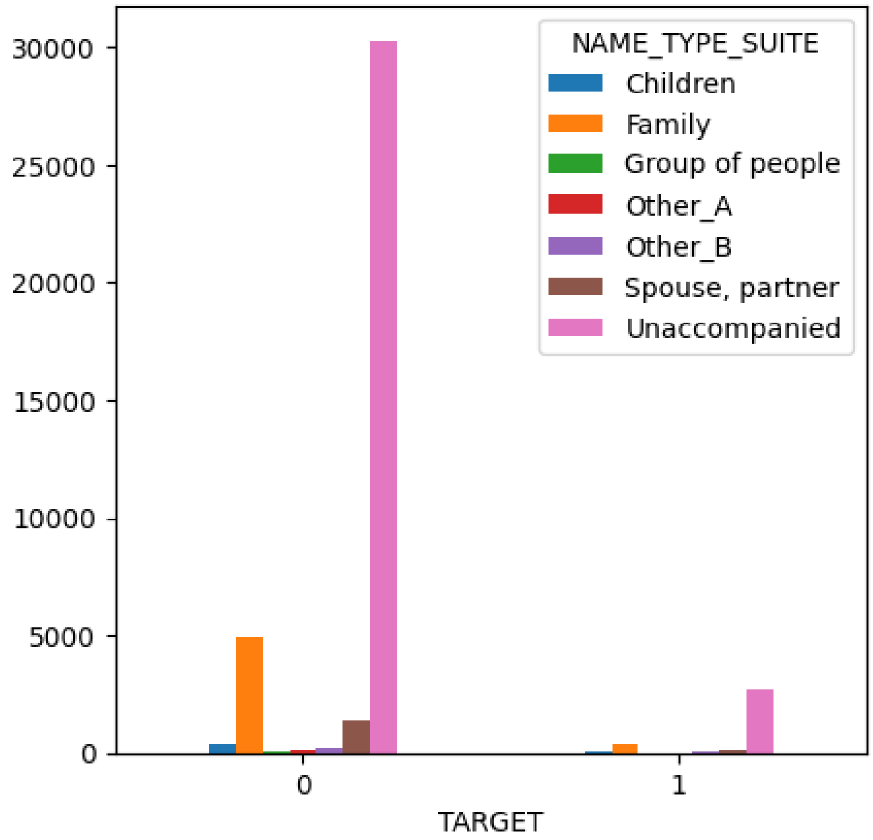


```
table_2=
pd.crosstab(index=application_data['TARGET'],columns=application_data[
'CODE_GENDER'])
print(table_2)
table_2.plot(kind="bar", figsize=(5,5),stacked=False)
plt.xticks(rotation=0)
plt.show()
#Females take more loans
```

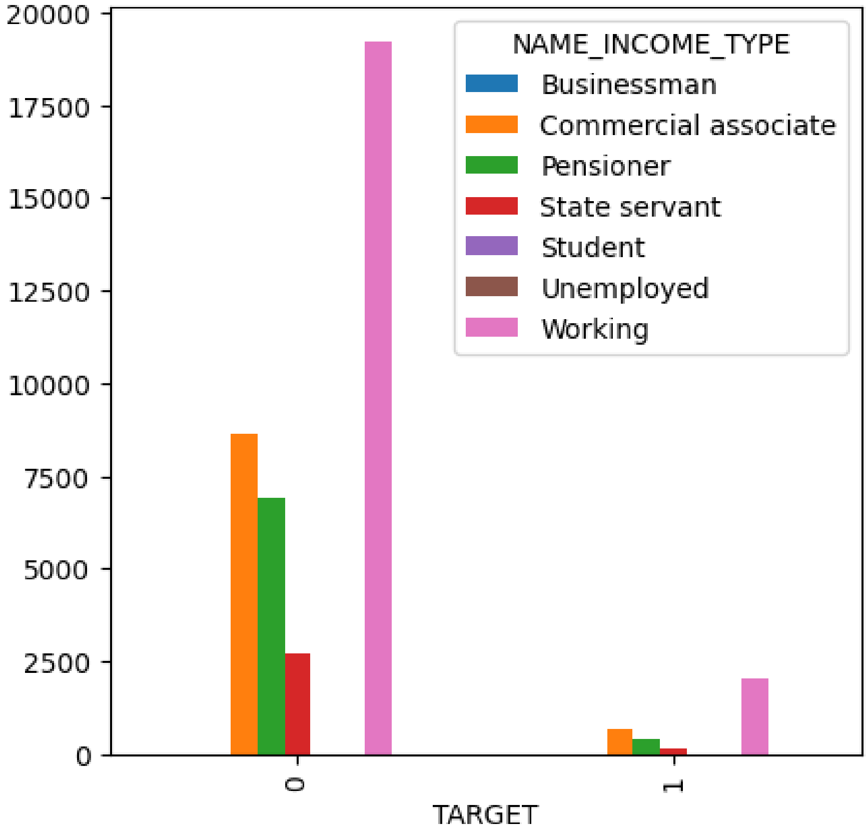
```
CODE_GENDER       F      M  XNA
TARGET
0            188278  94404    4
1             14170  10655    0
```



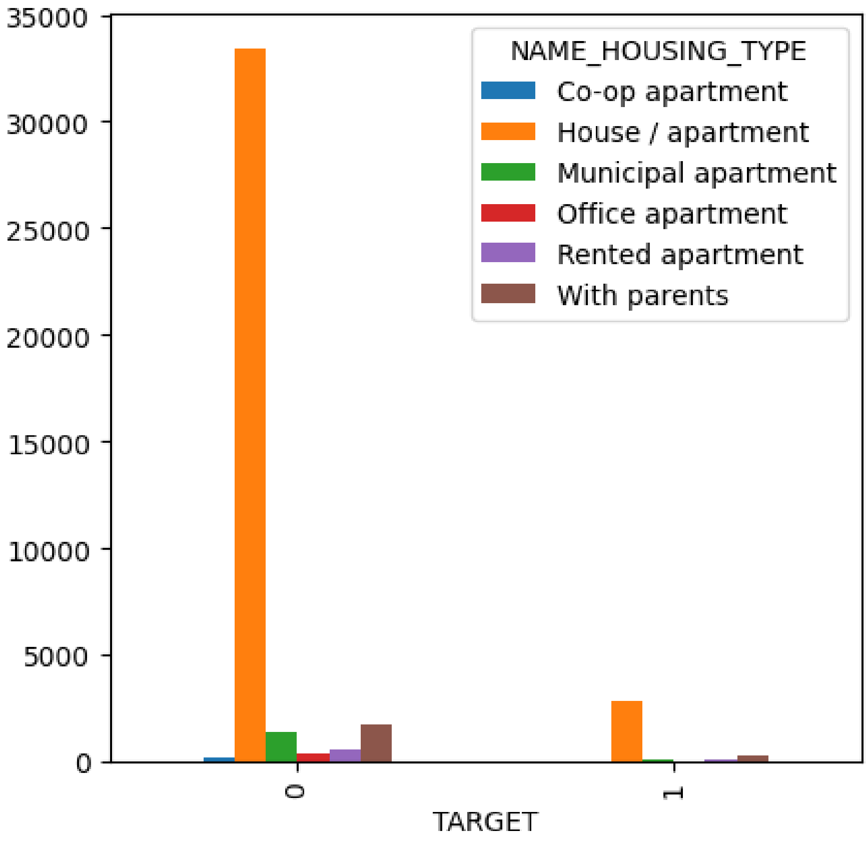
```
table_3=
pd.crosstab(index=application_data['TARGET'],columns=application_data[
'NAME_TYPE_SUITE'])
print(table_3)
table_3.plot(kind="bar", figsize=(5,5),stacked=False)
plt.xticks(rotation=0)
plt.show()
# Most of the people come alone when taking a loan

NAME_TYPE_SUITE  Children  Family  Group of people  Other_A
Other_B  \
TARGET

0                     404    4958               31      105      189

1                      40     409                1        8       23


NAME_TYPE_SUITE  Spouse, partner  Unaccompanied
TARGET
0                           1377          30266
1                            111           2703
```



```
table_4=
pd.crosstab(index=application_data['TARGET'],columns=application_data[
'NAME_INCOME_TYPE'])
print(table_4)
table_4.plot(kind="bar", figsize=(5,5),stacked=False)
plt.show()
# working people take more loans

NAME_INCOME_TYPE  Businessman  Commercial associate  Pensioner  State
servant  \
TARGET

0                           2                  8645       6906
2727
1                           0                   688        423
154

NAME_INCOME_TYPE  Student  Unemployed  Working
TARGET
0                       2           3    19198
1                       0           2     2032
```



```
table_5=
pd.crosstab(index=application_data['TARGET'],columns=application_data[
'NAME_HOUSING_TYPE'])
print(table_5)
table_5.plot(kind="bar", figsize=(5,5),stacked=False)
plt.show()
# People having house/appartment tend to take more loans

NAME_HOUSING_TYPE  Co-op apartment  House / apartment  Municipal
apartment  \
TARGET

0                              140              33393
1382
1                               11               2848
116

NAME_HOUSING_TYPE  Office apartment  Rented apartment  With parents
TARGET
0                               313               543          1712
1                                24                70           230
```

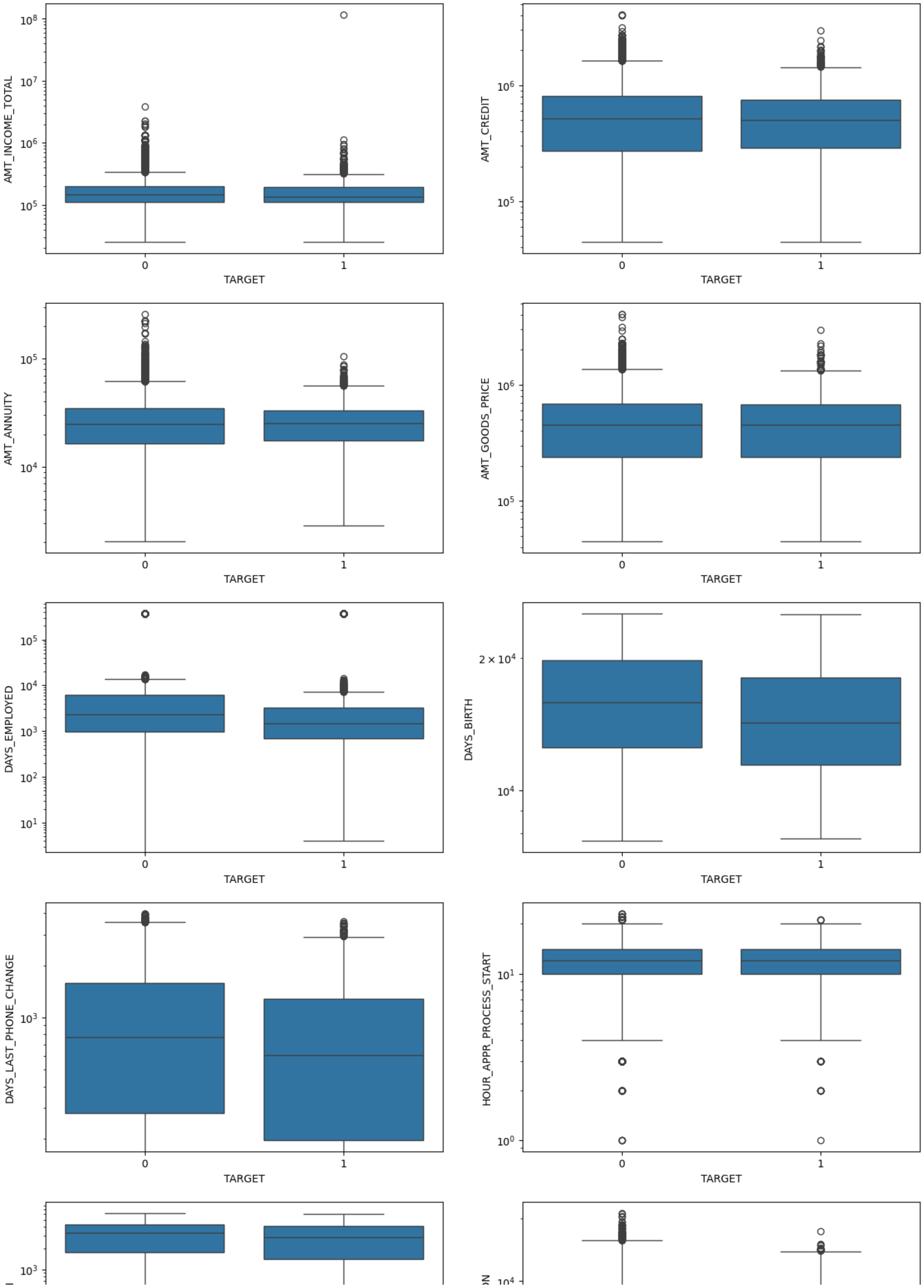


Bi-variate continous plots

```
application_data.head()
```

{"type":"dataframe","variable_name":"application_data"}

```
#Bi-variate continous plots
continous_columns=['AMT_INCOME_TOTAL','AMT_CREDIT','AMT_ANNUITY','AMT_
GOODS_PRICE',

'DAYS_EMPLOYED','DAYS_BIRTH','DAYS_LAST_PHONE_CHANGE','HOUR_APPR_PROCE
SS_START',
                  'DAYS_ID_PUBLISH','DAYS_REGISTRATION']
plt.figure(figsize=(15,25))
for i in (enumerate(continous_columns)):
    plt.subplot(len(continous_columns)//2,2,i[0]+1)

sns.boxplot(x='TARGET',y=application_data[i[1]].dropna(),data=applicat
ion_data)
    plt.yscale('log')
plt.show()
```


AMT_INCOME_TOTAL
$10^8$
$10^7$
$10^6$
$10^5$
0
1
TARGET
AMT_CREDIT
$10^6$
$10^5$
0
1
TARGET
AMT_ANNUITY
$10^5$
$10^4$
0
1
TARGET
AMT_GOODS_PRICE
$10^6$
$10^5$
0
1
TARGET
DAYS_EMPLOYED
$10^5$
$10^4$
$10^3$
$10^2$
$10^1$
0
1
TARGET
DAYS_BIRTH
$2 \times 10^4$
$10^4$
0
1
TARGET
DAYS_LAST_PHONE_CHANGE
$10^3$
0
1
TARGET
HOUR_APPR_PROCESS_START
$10^1$
$10^0$
0
1
TARGET
$10^3$
$10^4$

Insights

- There exists more clients who changed their their registration details after 4000 days of approval of loan.
- For few not default clients, time taken to publish id's are higher than default clients.
- The application process start hours taken for default and not default cases are similar.
- In non default cases, people keep their phone numbers for greater time.
- People with greater number of days born count are less likely to default.
- In non default case AMT_GOODS PRICE contains more outlers than default case.
- In default case, most of the clients amount annuity tends to be greater than 25000(median value).
- Whose credit amount is greater than 50000 tends to be less default than compared to default cases and vice versa.
- people with higher no of employment days are less likely to default.
- Majority of defaulting people are having less total income.

## Reading Previous application data

```
previous_data=pd.read_csv('previous_application.csv')
previous_data.head()
```

```
{"type":"dataframe","variable_name":"previous_data"}
```

```
# checking of missing values percentage
round((100*previous_data.isnull().sum()/len(previous_data)),2)
```

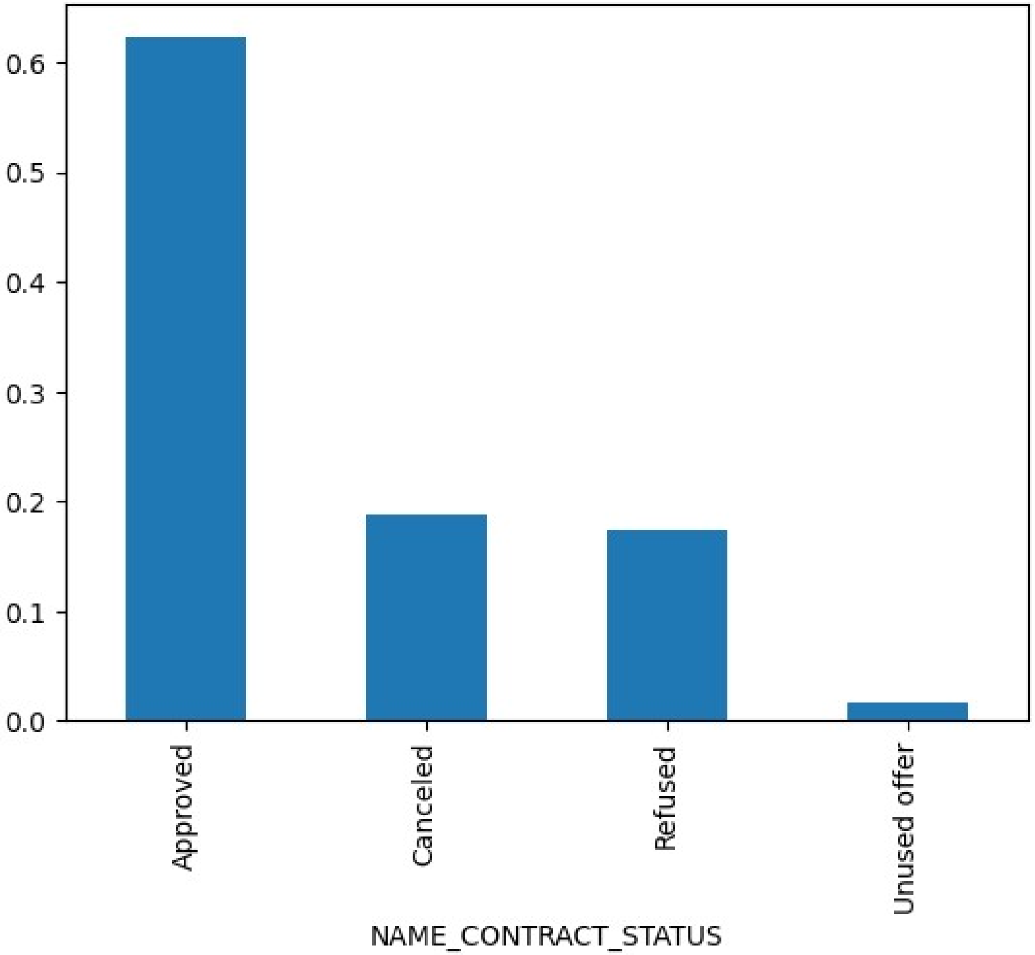
```
SK_ID_PREV                      0.00
SK_ID_CURR                      0.00
NAME_CONTRACT_TYPE              0.00
AMT_ANNUITY                    22.18
AMT_APPLICATION                 0.00
AMT_CREDIT                      0.00
AMT_DOWN_PAYMENT               53.19
AMT_GOODS_PRICE                22.93
WEEKDAY_APPR_PROCESS_START      0.00
HOUR_APPR_PROCESS_START         0.00
FLAG_LAST_APPL_PER_CONTRACT     0.00
NFLAG_LAST_APPL_IN_DAY          0.00
RATE_DOWN_PAYMENT              53.19
RATE_INTEREST_PRIMARY          99.65
RATE_INTEREST_PRIVILEGED       99.65
NAME_CASH_LOAN_PURPOSE          0.00
NAME_CONTRACT_STATUS            0.00
DAYS_DECISION                   0.00
NAME_PAYMENT_TYPE               0.00
CODE_REJECT_REASON              0.00
NAME_TYPE_SUITE                49.11
NAME_CLIENT_TYPE                0.00
NAME_GOODS_CATEGORY             0.00
NAME_PORTFOLIO                  0.00
```

```
NAME_PRODUCT_TYPE               0.00
CHANNEL_TYPE                    0.00
SELLERPLACE_AREA                0.00
NAME_SELLER_INDUSTRY            0.00
CNT_PAYMENT                    22.18
NAME_YIELD_GROUP                0.00
PRODUCT_COMBINATION             0.02
DAYS_FIRST_DRAWING             40.08
DAYS_FIRST_DUE                 40.08
DAYS_LAST_DUE_1ST_VERSION      40.08
DAYS_LAST_DUE                  40.08
DAYS_TERMINATION               40.08
NFLAG_INSURED_ON_APPROVAL      40.08
dtype: float64
```

```
# removing those columns which are having null percentage greater than
50
#
AMT_DOWN_PAYMENT,RATE_DOWN_PAYMENT,RATE_INTEREST_PRIMARY,RATE_INTEREST
_PRIVILEGED
previous_data=previous_data.drop(['AMT_DOWN_PAYMENT','RATE_DOWN_PAYMEN
T','RATE_INTEREST_PRIMARY','RATE_INTEREST_PRIVILEGED'], axis = 1)
previous_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 893889 entries, 0 to 893888
Data columns (total 33 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   SK_ID_PREV                   893889 non-null  int64
 1   SK_ID_CURR                   893889 non-null  int64
 2   NAME_CONTRACT_TYPE           893888 non-null  object
 3   AMT_ANNUITY                  695585 non-null  float64
 4   AMT_APPLICATION              893888 non-null  float64
 5   AMT_CREDIT                   893888 non-null  float64
 6   AMT_GOODS_PRICE              688879 non-null  float64
 7   WEEKDAY_APPR_PROCESS_START   893888 non-null  object
 8   HOUR_APPR_PROCESS_START      893888 non-null  float64
 9   FLAG_LAST_APPL_PER_CONTRACT  893888 non-null  object
 10  NFLAG_LAST_APPL_IN_DAY       893888 non-null  float64
 11  NAME_CASH_LOAN_PURPOSE       893888 non-null  object
 12  NAME_CONTRACT_STATUS         893888 non-null  object
 13  DAYS_DECISION                893888 non-null  float64
 14  NAME_PAYMENT_TYPE            893888 non-null  object
 15  CODE_REJECT_REASON           893888 non-null  object
 16  NAME_TYPE_SUITE              454865 non-null  object
 17  NAME_CLIENT_TYPE             893888 non-null  object
 18  NAME_GOODS_CATEGORY          893888 non-null  object
 19  NAME_PORTFOLIO               893888 non-null  object
 20  NAME_PRODUCT_TYPE            893888 non-null  object
 21  CHANNEL_TYPE                 893888 non-null  object 
 22  SELLERPLACE_AREA             893888 non-null  float64
 23  NAME_SELLER_INDUSTRY         893888 non-null  object 
 24  CNT_PAYMENT                  695588 non-null  float64
 25  NAME_YIELD_GROUP             893888 non-null  object 
 26  PRODUCT_COMBINATION          893709 non-null  object 
 27  DAYS_FIRST_DRAWING           535652 non-null  float64
 28  DAYS_FIRST_DUE               535652 non-null  float64
 29  DAYS_LAST_DUE_1ST_VERSION    535652 non-null  float64
 30  DAYS_LAST_DUE                535652 non-null  float64
 31  DAYS_TERMINATION             535652 non-null  float64
 32  NFLAG_INSURED_ON_APPROVAL    535652 non-null  float64
dtypes: float64(15), int64(2), object(16)
memory usage: 225.1+ MB
```

```
# converting -ve values to +ve
previous_data['DAYS_DECISION']=previous_data['DAYS_DECISION'].abs()
previous_data['SELLERPLACE_AREA']=previous_data['SELLERPLACE_AREA'].abs()
previous_data['DAYS_FIRST_DUE']=previous_data['DAYS_FIRST_DUE'].abs()
previous_data['DAYS_LAST_DUE_1ST_VERSION']=previous_data['DAYS_LAST_DUE_1ST_VERSION'].abs()
previous_data['DAYS_LAST_DUE']=previous_data['DAYS_LAST_DUE'].abs()
previous_data['DAYS_TERMINATION']=previous_data['DAYS_TERMINATION'].abs()
previous_data['DAYS_FIRST_DRAWING']=previous_data['DAYS_FIRST_DRAWING'].abs()

(previous_data.NAME_CONTRACT_STATUS.value_counts()/len(previous_data)).plot.bar()
plt.show()
```



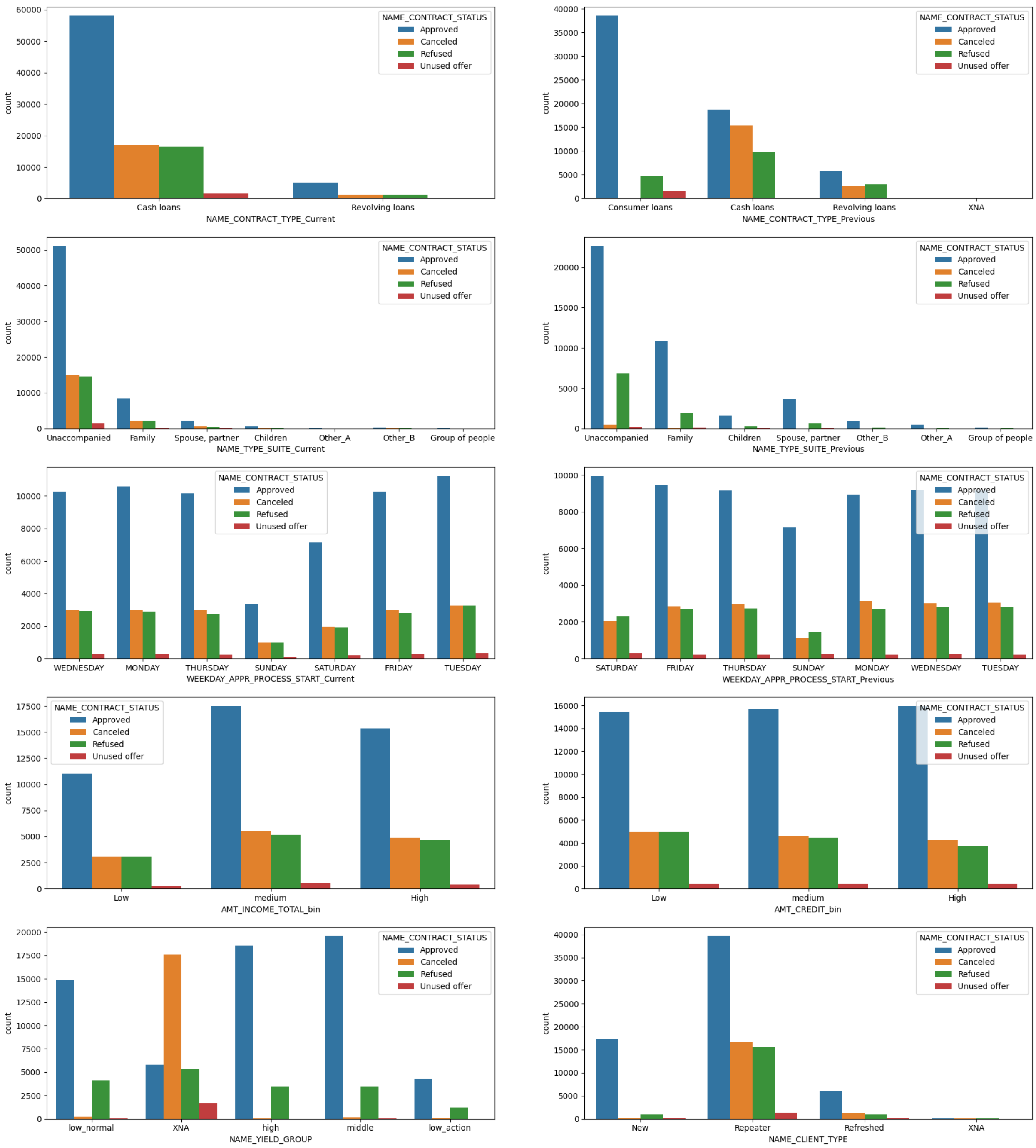
Merging application data and previous application data

```
# making a left join because we need all the rows in application data
# by making this left join we get historical application data for each
applicant.
# if we made a inner join we would loose the data of a new customer
who doesn't have a previous record.
# Current data will get duplicated the exact number of times it is
found in previous application data.
# with this in mind we are moving forward.

merged_df=pd.merge(application_data,previous_data,how='left',on='SK_ID
_CURR',suffixes=('_Current', '_Previous'))
merged_df.head()

{"type":"dataframe","variable_name":"merged_df"}
```

Univariate Analysis

```
Categorical analysis
```

```
# Univariate Categorical analysis
categorical_columns=['NAME_CONTRACT_TYPE_Current','NAME_CONTRACT_TYPE_
Previous',

'NAME_TYPE_SUITE_Current','NAME_TYPE_SUITE_Previous',

'WEEKDAY_APPR_PROCESS_START_Current','WEEKDAY_APPR_PROCESS_START_Previ
ous',

'AMT_INCOME_TOTAL_bin','AMT_CREDIT_bin','NAME_YIELD_GROUP','NAME_CLIEN
T_TYPE']


plt.figure(figsize=(22,25))
for i in (enumerate(categorical_columns)):
    plt.subplot(len(categorical_columns)//2,2,i[0]+1)
    sns.countplot(x=i[1],hue='NAME_CONTRACT_STATUS',data=merged_df)
    #lt.yscale('log')
    #plt.xticks(rotation=90)
plt.show()
```



## Insights

- Repeater has highest number of approved loans.
- Middle NAME_YIELD_GROUP has highest approval.
- Value of AMT_CREDIT_BIN does not affect loan approvals.
- for Medium AMT_INCOME_TOTAL_bin the approval is highest .
- in previous application saturday has the highest approval rate.
- but in current application it is tuesday.

- both in NAME_CONTRACT_TYPE_Previous and NAME_CONTRACT_TYPE_Current unaccompanied has the highest number.
- currently bank is only giving two types of loans -Cash and Revolving Loans.
- Previously bank was providing Cash, Revolving and Consumer loans.
- Number of consumer loans were highest previously and now highest number is Cash loans.

# Continous/Numerical analysis

```
# Univariate Numerical analysis
continous_columns=['AMT_CREDIT_Previous','AMT_CREDIT_Current','AMT_ANN
UITY_Current','AMT_ANNUITY_Previous',

'AMT_GOODS_PRICE_Current','AMT_GOODS_PRICE_Previous','CNT_FAM_MEMBERS'
,'CNT_CHILDREN',

'HOUR_APPR_PROCESS_START_Previous','HOUR_APPR_PROCESS_START_Current']
plt.figure(figsize=(22,25))
for i in (enumerate(continous_columns)):
    plt.subplot(len(continous_columns)//2,2,i[0]+1)

sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')

sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})

sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
    # we added kde_kws={'bw':0.1} in parameter to overcome bandwidth
limitation.
    sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')

plt.show()
```

```
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
```

```
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
```

```
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
```

```
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
```

```
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
```

```
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.
```

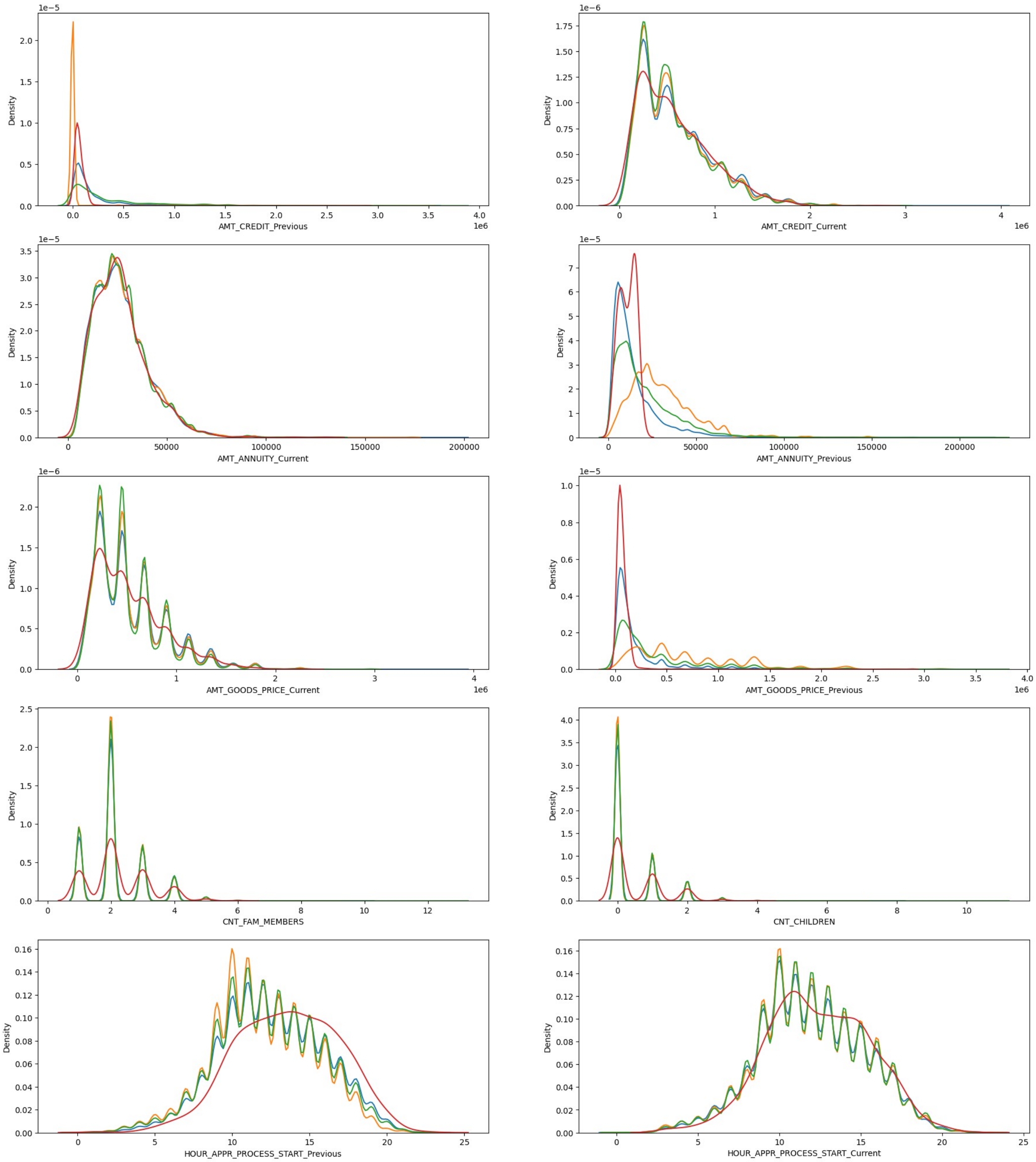
```
Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.
```

```
  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
```

```
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
<ipython-input-74-d2e9b5a7231d>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Approved',
:][i[1]].dropna(),hist=False,label='Approved')
<ipython-input-74-d2e9b5a7231d>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
```

```
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Canceled',
:][i[1]].dropna(),hist=False,label='Canceled',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751


sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Refused',:
][i[1]].dropna(),hist=False,label='Refused',kde_kws={'bw':0.1})
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2496:
UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and
`bw_adjust`.
Setting `bw_method=0.1`, but please see the docs for the new
parameters
and update your code. This will become an error in seaborn v0.14.0.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-74-d2e9b5a7231d>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).
```

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(merged_df.loc[merged_df.NAME_CONTRACT_STATUS=='Unused
offer',:][i[1]].dropna(),hist=False,label='Unused offer')
```



## Insights

As we can see from graphs

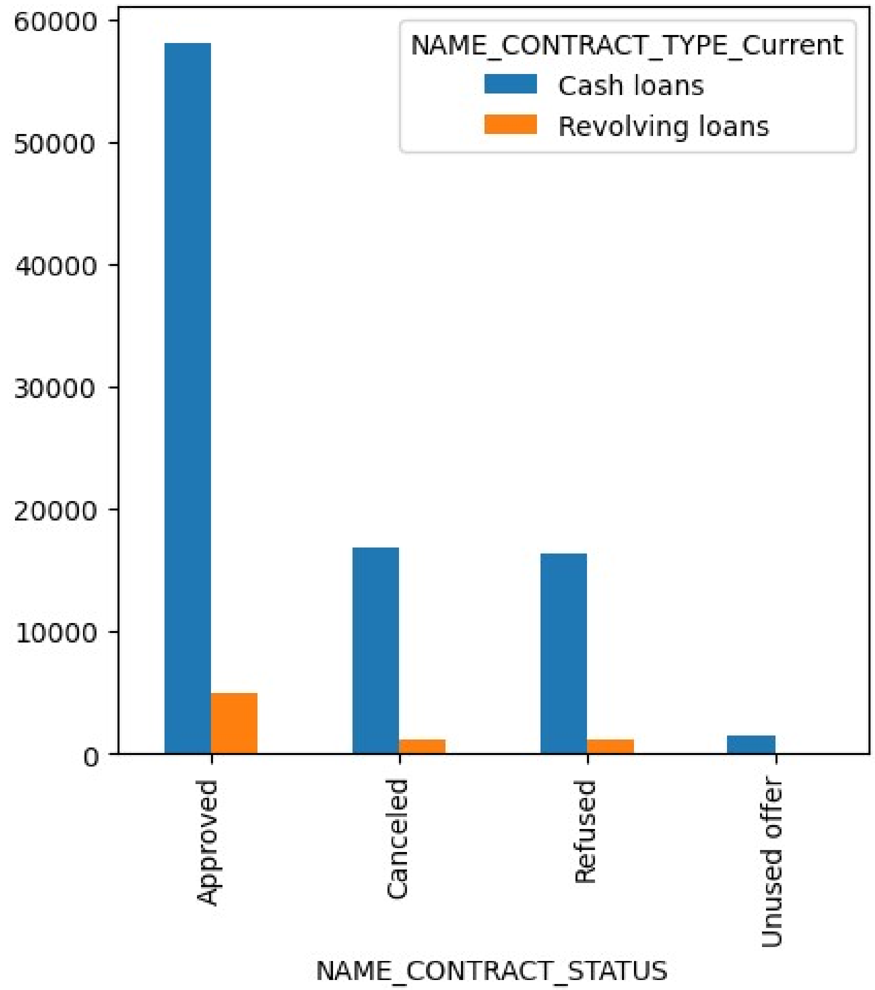
- High number of applications are filed in 9 AM to 2 PM for both Current and Previous data.
- So busiest hours for bank are form 9 AM to 2 PM.
- nuclear family tends to take more loans.
- Previously bank had high unused offers but currently refused is high incase of AMT_GOODS_PRICE.
- Previously bank had high unused offers and currently cancelled/refused offers are similar for AMT_ANNUITY.
- Previously bank had high unused offers and currently high number of refused offers for AMT_CREDIT.

Bi-variate Analysis

#Categorical

```
table_6=
pd.crosstab(index=merged_df['NAME_CONTRACT_STATUS'],columns=merged_df[
'NAME_CONTRACT_TYPE_Current'])
print(table_6)
table_6.plot(kind="bar", figsize=(5,5),stacked=False)
plt.show()
#Cash loans have the highest count of Approved loans
```

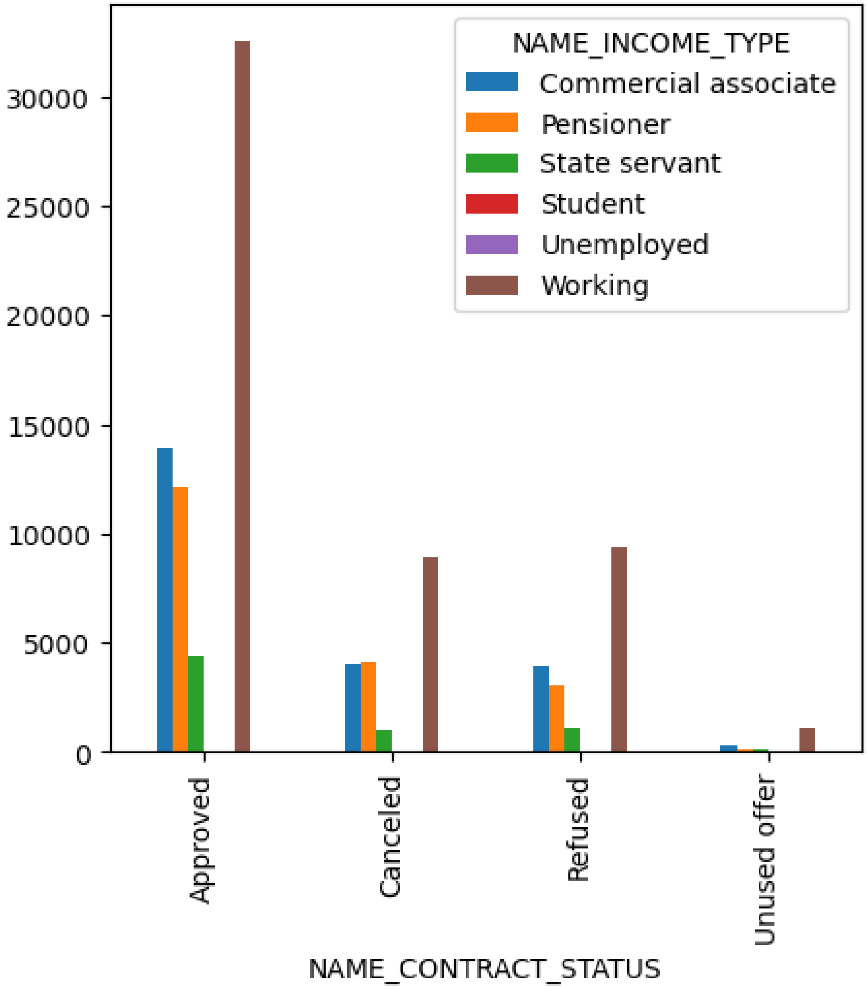
```
NAME_CONTRACT_TYPE_Current  Cash loans  Revolving loans
NAME_CONTRACT_STATUS
Approved                         58057             4983
Canceled                         16937             1225
Refused                          16357             1160
Unused offer                      1532              145
```



```
table_9=
pd.crosstab(index=merged_df['NAME_CONTRACT_STATUS'],columns=merged_df[
'NAME_INCOME_TYPE'])
print(table_9)
table_9.plot(kind="bar", figsize=(5,5),stacked=False)
plt.show()
# Highest number of approvals for working applicant
```

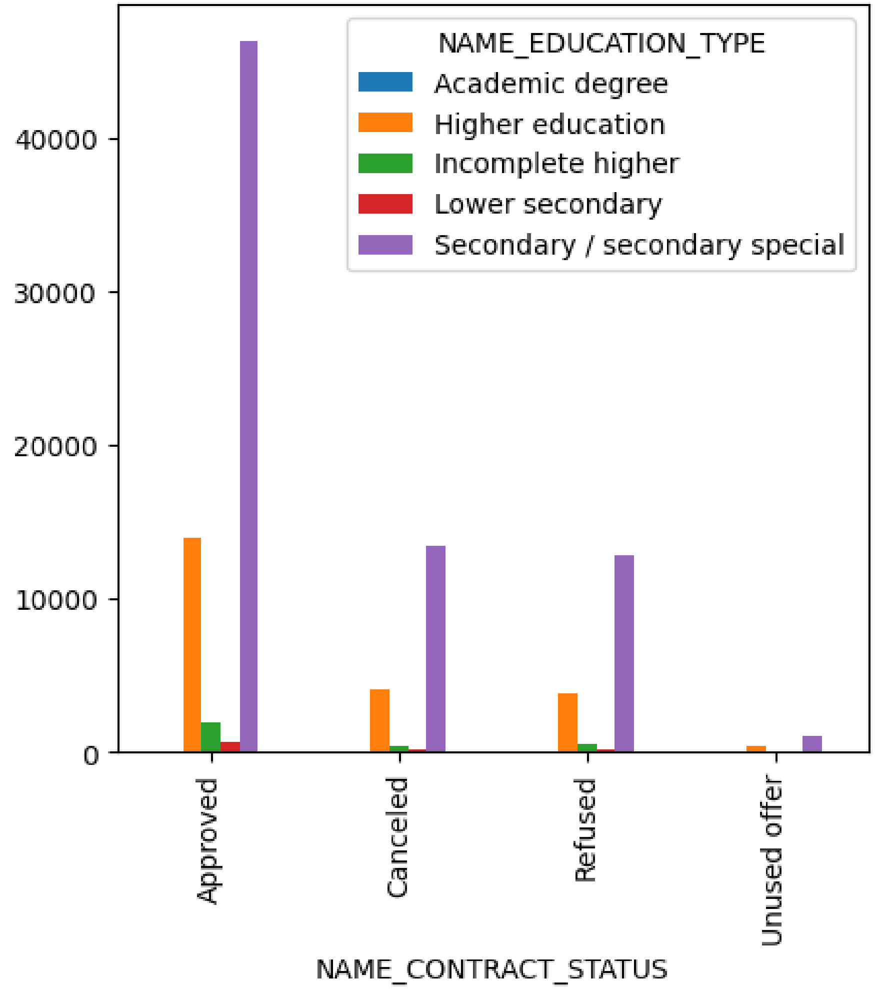
```
NAME_INCOME_TYPE      Commercial associate  Pensioner  State servant
Student  \
NAME_CONTRACT_STATUS

Approved                             13880      12143           4417
2
Canceled                              4053       4101           1063
0
Refused                               3962       3074           1094
0
Unused offer                           353        143            113
```

```
0

NAME_INCOME_TYPE      Unemployed  Working
NAME_CONTRACT_STATUS
Approved                      12    32586
Canceled                       1     8944
Refused                       11     9376
Unused offer                   0     1068
```



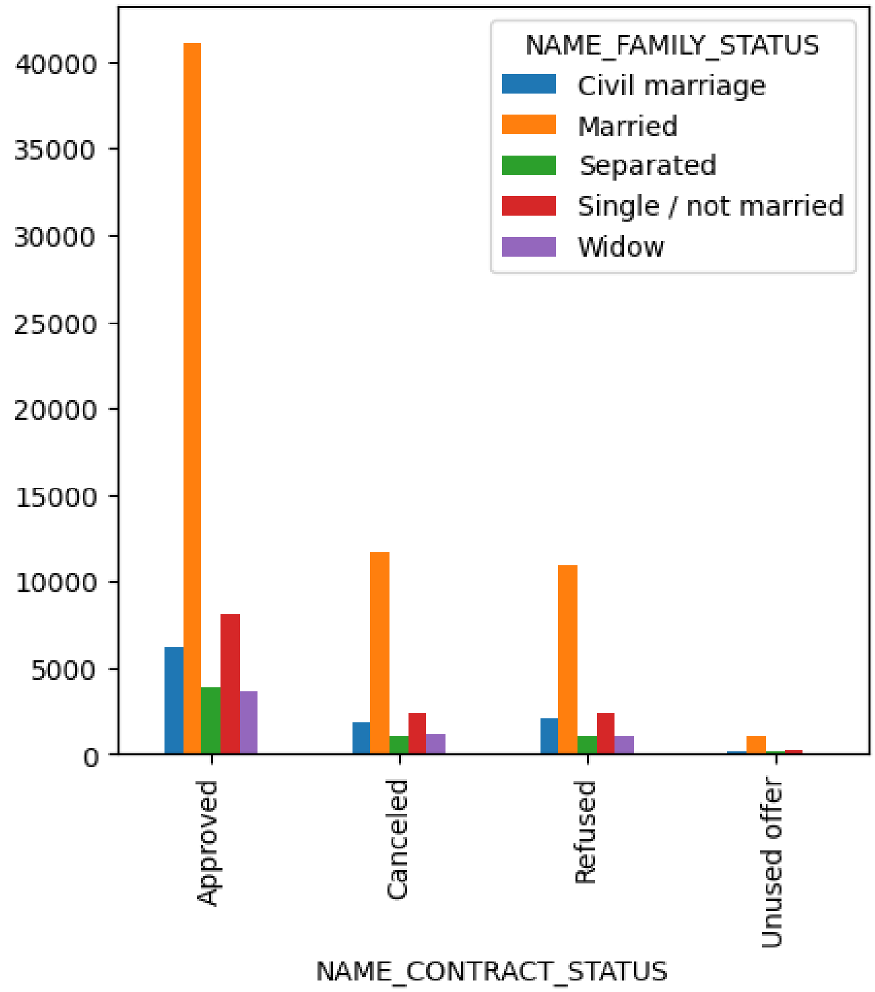
```
table_10=
pd.crosstab(index=merged_df['NAME_CONTRACT_STATUS'],columns=merged_df[
'NAME_EDUCATION_TYPE'])
print(table_10)
table_10.plot(kind="bar", figsize=(5,5),stacked=False)
plt.show()
# Highest number of approvals for Secondary/secondary special educated
applicant
```

```
NAME_EDUCATION_TYPE   Academic degree  Higher education  Incomplete
higher  \
NAME_CONTRACT_STATUS

Approved                           18             13990
1944
Canceled                            0              4099
462
Refused                             7              3800
617
Unused offer                        0               451
86

NAME_EDUCATION_TYPE   Lower secondary  Secondary / secondary special
NAME_CONTRACT_STATUS
Approved                          741                          46347
Canceled                          200                          13401
Refused                           221                          12872
Unused offer                       11                           1129
```



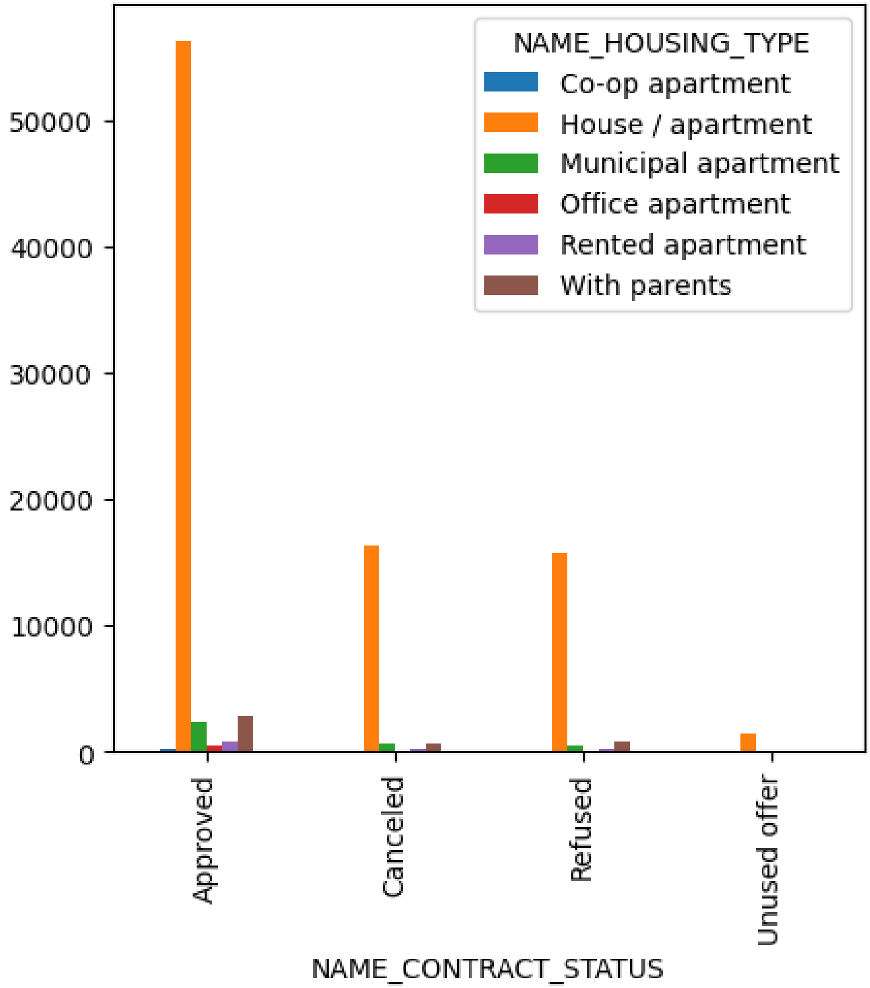
```
table_11=
pd.crosstab(index=merged_df['NAME_CONTRACT_STATUS'],columns=merged_df[
'NAME_FAMILY_STATUS'])
print(table_11)
table_11.plot(kind="bar", figsize=(5,5),stacked=False)
plt.show()
# Highest number of approvals for Married applicant
```

```
NAME_FAMILY_STATUS    Civil marriage  Married  Separated  \
NAME_CONTRACT_STATUS
Approved                        6278    41131       3888
Canceled                        1882    11732       1053
Refused                         2026    10882       1089
Unused offer                     143     1062        127

NAME_FAMILY_STATUS    Single / not married  Widow
NAME_CONTRACT_STATUS
Approved                              8093   3650
Canceled                              2373   1122
Refused                               2445   1075
Unused offer                           303     42
```



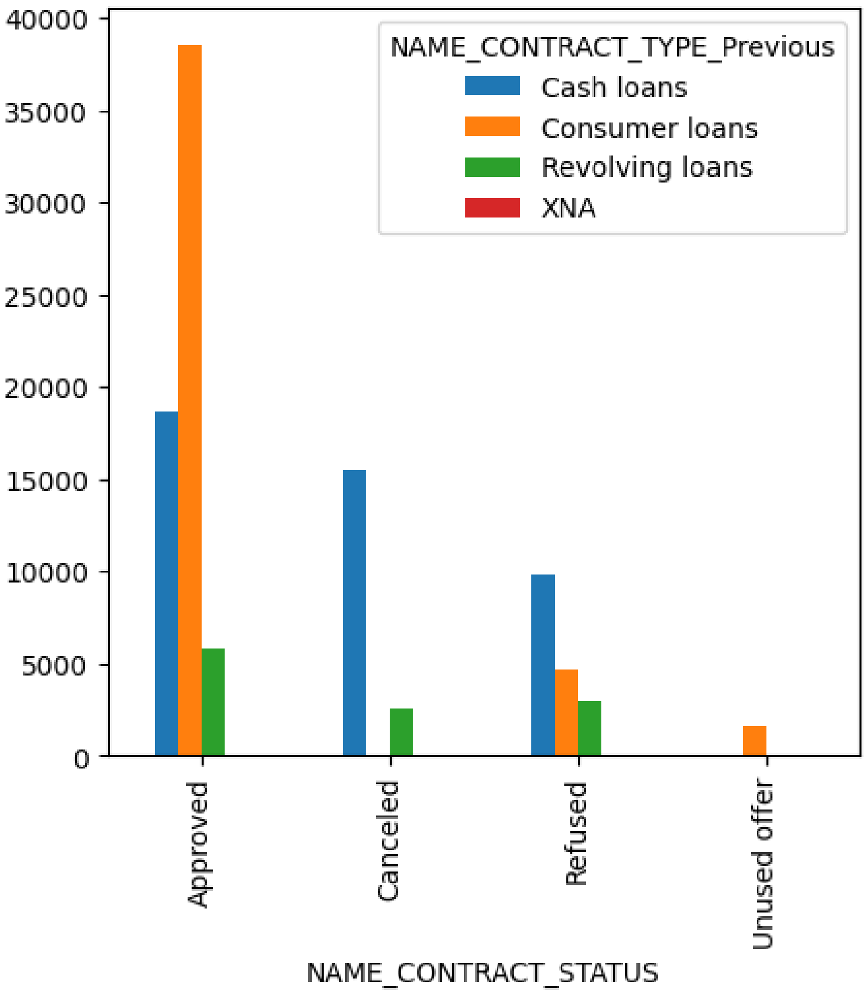
```
table_12=
pd.crosstab(index=merged_df['NAME_CONTRACT_STATUS'],columns=merged_df[
'NAME_HOUSING_TYPE'])
print(table_12)
table_12.plot(kind="bar", figsize=(5,5),stacked=False)
plt.show()
# Highest number of approvals for House/apartment owner.

NAME_HOUSING_TYPE      Co-op apartment  House / apartment  Municipal
apartment  \
NAME_CONTRACT_STATUS

Approved                           181              56388
2396
Canceled                            36              16412
636
Refused                             36              15726
577
Unused offer                        27               1470
```

```
59
NAME_HOUSING_TYPE     Office apartment  Rented apartment  With parents
NAME_CONTRACT_STATUS
Approved                           464               847          2764
Canceled                           128               197           753
Refused                            135               240           803
Unused offer                        10                16            95
```



```
table_15=
pd.crosstab(index=merged_df['NAME_CONTRACT_STATUS'],columns=merged_df[
'NAME_CONTRACT_TYPE_Previous'])
print(table_15)
table_15.plot(kind="bar", figsize=(5,5),stacked=False)
```

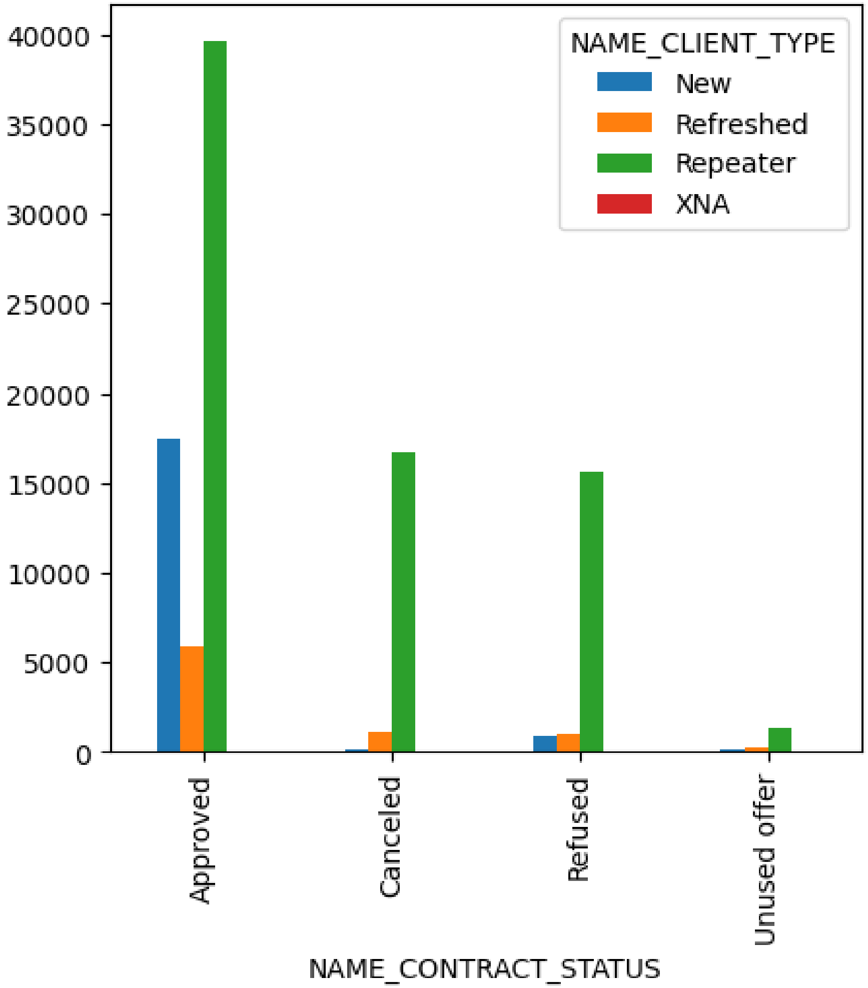
```
plt.show()
# Highest number of approvals for Consumer Loans.

NAME_CONTRACT_TYPE_Previous  Cash loans  Consumer loans  Revolving
loans  XNA
NAME_CONTRACT_STATUS

Approved                          18702           38544
5794    0
Canceled                          15469             101
2572   20
Refused                            9850            4646
3020    1
Unused offer                         31            1646
0    0
```



```
table_17=
pd.crosstab(index=merged_df['NAME_CONTRACT_STATUS'],columns=merged_df[
'NAME_CLIENT_TYPE'])
```

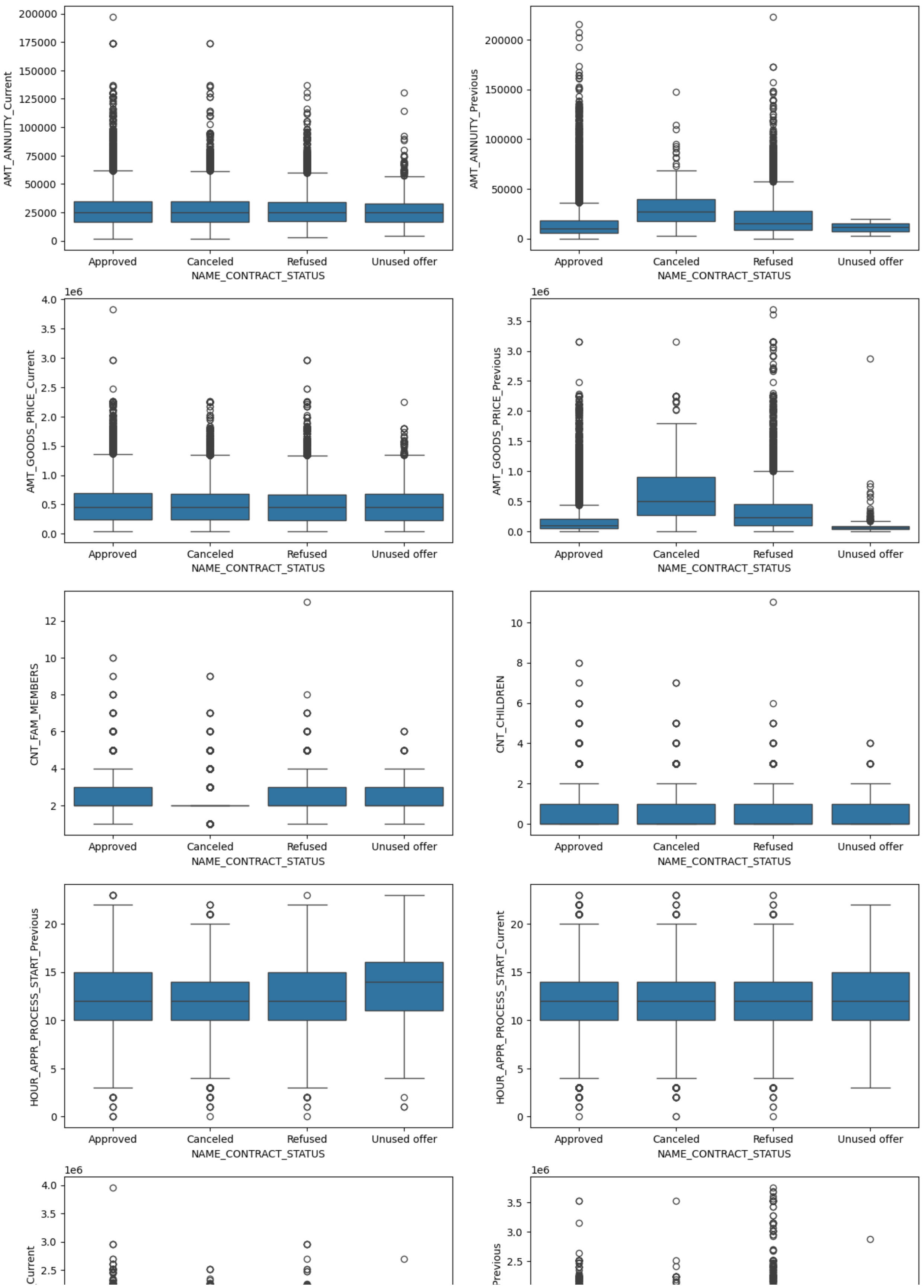
```
print(table_17)
table_17.plot(kind="bar", figsize=(5,5),stacked=False)
plt.show()
# repeated applications got approved most number of times
```

```
NAME_CLIENT_TYPE        New  Refreshed  Repeater  XNA
NAME_CONTRACT_STATUS
Approved              17425       5934     39651   30
Canceled                198       1186     16725   53
Refused                 904        975     15617   21
Unused offer            136        219      1320    2
```



# Continous/Numerical analysis

```
#Bi-variate continous plots
continous_columns=['AMT_ANNUITY_Current','AMT_ANNUITY_Previous',

'AMT_GOODS_PRICE_Current','AMT_GOODS_PRICE_Previous','CNT_FAM_MEMBERS'
,'CNT_CHILDREN',
'HOUR_APPR_PROCESS_START_Previous','HOUR_APPR_PROCESS_START_Current',
                   'AMT_CREDIT_Current','AMT_CREDIT_Previous']
                   #'AMT_INCOME_TOTAL']
plt.figure(figsize=(15,25))
for i in (enumerate(continous_columns)):
    plt.subplot(len(continous_columns)//2,2,i[0]+1)

sns.boxplot(x='NAME_CONTRACT_STATUS',y=merged_df[i[1]].dropna(),data=m
erged_df)
plt.show()
```


AMT_ANNUITY_Current
200000
175000
150000
125000
100000
75000
50000
25000
0
Approved
Canceled
Refused
Unused offer
NAME_CONTRACT_STATUS
AMT_ANNUITY_Previous
200000
150000
100000
50000
0
Approved
Canceled
Refused
Unused offer
NAME_CONTRACT_STATUS
1e6
AMT_GOODS_PRICE_Current
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Approved
Canceled
Refused
Unused offer
NAME_CONTRACT_STATUS
1e6
AMT_GOODS_PRICE_Previous
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Approved
Canceled
Refused
Unused offer
NAME_CONTRACT_STATUS
CNT_FAM_MEMBERS
12
10
8
6
4
2
Approved
Canceled
Refused
Unused offer
NAME_CONTRACT_STATUS
CNT_CHILDREN
10
8
6
4
2
0
Approved
Canceled
Refused
Unused offer
NAME_CONTRACT_STATUS
HOUR_APPR_PROCESS_START_Previous
20
15
10
5
0
Approved
Canceled
Refused
Unused offer
NAME_CONTRACT_STATUS
HOUR_APPR_PROCESS_START_Current
20
15
10
5
0
Approved
Canceled
Refused
Unused offer
NAME_CONTRACT_STATUS
1e6
Current
4.0
3.5
3.0
2.5
1e6
Previous
3.5
3.0
2.5

### Insights

- AMT_CREDIT_Previous has highest refused cases and AMT_CREDIT_Current is similar for all 4 cases.
- time spent in unused offer is higher as compared to other categories.
- So bank should reduce time spent on unused offer.
- nuclear family(2-3 people in family) get highest approval.
- Previously most of the applications were cancelled or refused
- but now Refused/Cancelled/Approved/Unused all four have similar situation for AMT_GOODS_PRICE.
- Previously most of the applications were cancelled or refused
- but now Refused/Cancelled/Approved/Unused all four have similar situation for AMT_ANNUITY.

## Final Words

Target/focused variable for Application dataset - **TARGET** Target/focused variable for Previous dataset - **NAME_CONTRACT_STATUS**

Top Major variables to consider for loan prediction:

1. NAME_EDUCATION_TYPE
2. AMT_INCOME_TOTAL
3. DAYS_BIRTH
4. AMT_CREDIT
5. DAYS_EMPLOYED
6. AMT_ANNUITY
7. NAME_INCOME_TYPE
8. CODE_GENDER
9. NAME_HOUSING_TYPE

The above mentioned variables are to be considered before approving application to minimize risk of loss.